

TUGAS AKHIR – SM141501

**DESAIN KENDALI HALUAN KAPAL SSV BRP
TARLAC (LD-601) MENGGUNAKAN ADAPTIVE PID
CONTROLLER**

**MOCH. ARDI FIRMANSYAH
NRP. 1213 100 073**

**Dosen Pembimbing :
Dr. Didik Khusnul Arif, S.Si, M.Si
Tahiyatul Asfihani, S.Si, M.Si**

**JURUSAN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017**



TUGAS AKHIR - SM141501

**DESAIN KENDALI HALUAN KAPAL SSV BRP
TARLAC (LD-601) MENGGUNAKAN ADAPTIVE PID
CONTROLLER**

MOCH. ARDI FIRMANSYAH
NRP 1213 100 073

Pembimbing:
Dr. Didik Khusnul Arif, S.Si, M.Si
Tahiyatul Asfihani, S.Si, M.Si

JURUSAN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017

Halaman ini sengaja dikosongkan



FINAL PROJECT - SM141501

**CONTROLLER DESIGN FOR SSV BRP TARLAC
(LD-601) SHIP HEADING USING ADAPTIVE PID
CONTROLLER**

MOCH. ARDI FIRMANSYAH
NRP 1213 100 073

Supervisors:
Dr. Didik Khusnul Arif, S.Si, M.Si
Tahiyatul Asfihani, S.Si, M.Si

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Natural Sciences
Sepuluh Nopember Institute of Technology
Surabaya 2017

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

**DESAIN SISTEM KENDALI ADAPTIVE PID UNTUK
HALUAN KAPAL SSV BRP TARLAC (LD-601)**

**SHIP HEADING CONTROL DESIGN FOR SSV BRP
TARLAC (LD-601) USING ADAPTIVE PID CONTROLLER**

Diajukan Untuk Memenuhi Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Sains
pada

Bidang Studi Matematika Terapan
Program Studi S-1 Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya
Oleh:

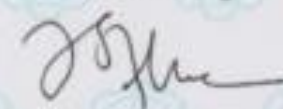
MOCH. ARDI FIRMANSYAH

NRP. 1213 100 073

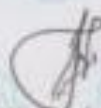
Menyetujui,

Pembimbing II

Pembimbing I



Tahiyatul Asfihani, S.Si, M.Si
NIP. 19870728 201404 2 001



Dr. Didik Khusnul Arif, S.Si, M.Si
NIP. 19730930 199702 1 001

Mengetahui,

Ketua Jurusan Matematika
FMIPA-ITS



Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003
Surabaya, 19 Januari 2017

Halaman ini sengaja dikosongkan

DESAIN KENDALI HALUAN KAPAL SSV BRP TARLAC (LD-601) MENGGUNAKAN ADAPTIVE PID CONTROLLER

Nama Mahasiswa : Moch. Ardi Firmansyah
NRP : 1213 100 073
Jurusan : Matematika FMIPA-ITS
Pembimbing : 1. Dr. Didik Khusnul Arif, S.Si, M.Si
2. Tahiyatul Asfihani, S.Si, M.Si

Abstrak

Kendali PID Adaptif dan PID konvensional diterapkan pada sistem autopilot kapal untuk mengendalikan gerak *yaw* pada kapal atau kendali haluan kapal SSV BRP TARLAC (LD-601). Model matematika yang digunakan berdasarkan pendekatan fungsi transfer model Nomoto orde 2 pada dinamika kapal SSV BRP TARLAC (LD-601). Selanjutnya, fungsi transfer akan diubah dalam persamaan ruang keadaan dengan menggunakan invers transformasi laplace. Desain kendali PID adaptif yang digunakan berdasarkan PID *Self Tuning Regulator* dengan parameter K_p , K_i , K_d pada pengendali PID dilakukan proses *online tuning* menggunakan metode *recursive least squares* sehingga parameter dapat menyesuaikan pada saat sistem mengalami perubahan. Simulasi dengan menggunakan MATLAB diberikan untuk membandingkan efektivitas dari PID konvensional dan PID adaptif yang diusulkan. Dari hasil simulasi diharapkan bahwa desain PID adaptif mampu menyesuaikan kendali secara *online* saat sistem mengalami gangguan. Dari hasil simulasi menunjukkan kendali PID adaptif mempunyai kinerja yang lebih baik dibandingkan dengan PID konvensional dengan $t_d = 0,3$, $t_p = 1,1$, $t_r = 1$, $M_p = 0,3\%$, $t_s = 1$. Pengendali PID adaptif mampu kembali menuju *setpoint* saat sistem mengalami gangguan dibandingkan PID konvensional.

Kata kunci: Kendali Adaptif, Kendali PID adaptif, *Recursive Least Squares*, Haluan Kapal, Kapal SSV BRP TARLAC (LD-601)

Halaman ini sengaja dikosongkan

SHIP HEADING CONTROL DESIGN FOR SSV BRP TARLAC (LD-601) USING ADAPTIVE PID CONTROLLER

Name : Moch. Ardi Firmansyah
NRP : 1213 100 073
Department : Matematika FMIPA-ITS
Supervisor : 1. Dr. Didik Khusnul Arif, S.Si, M.Si
2. Tahiyatul Asfihani, S.Si, M.Si

Abstract

Adaptive PID controller and conventional PID controller are adopted as autopilot system for a ship to control yaw motion or ship heading control of SSV BRP TARLAC (LD-601). Mathematical model is based on transfer function on Nomoto's second order of SSV BRP TARLAC (LD-601) ship dynamic. The transfer function will be change to state-space equation using inverse Laplace transform. Design of adaptive PID controller is based on PID Self Tuning Regulator with K_p , K_i , K_d is PID gain are tuned using recursive least squares (RLS) method to adaptable when system will be change. Simulation using MATLAB are given to compare the effectiveness of the proposed Adaptive PID controller and Conventional PID controller. From the simulation show that adaptive PID controller can change the gain regulator immediately when system has disturbance. From the simulation respon transient adaptive PID controller has $t_d = 0,3$ $t_p = 1,1$, $t_r = 1$, $M_p = 0,3\%$ $t_s = 1$

Keywords: Adaptive Controller, Adaptive PID controller, Recursive Least squares, Ship Heading, SSV BRP TARLAC (LD-601) ship

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Syukur alhamdulillah atas kehadiran Allah SWT yang telah melimpahkan segala nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

**“DESAIN KENDAI HALUAN KAPAL SSV BRP TARLAC
(LD-601) MENGGUNAKAN ADAPTIVE PID
CONTROLLER”**

sebagai salah satu syarat kelulusan Program Sarjana Jurusan Matematika FMIPA Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Tugas Akhir ini dapat terselesaikan dengan baik berkat bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih dan penghargaan kepada:

1. Bapak Dr. Imam Mukhlash, S.Si, MT sebagai Ketua Jurusan Matematika FMIPA-ITS.
2. Bapak Dr. Didik Khusnul Arif, S.Si, M.Si sebagai dosen pembimbing I yang telah memberikan motivasi dan pengarahan dalam penyelesaian Tugas Akhir ini.
3. Ibu Tahiyatul Asfihani, S.Si, M.Si sebagai dosen pembimbing II yang telah memberikan motivasi dan pengarahan dalam penyelesaian Tugas Akhir ini.
4. Ibu Sunarsini S.Si, M.Si sebagai dosen wali selama penulis kuliah di Jurusan Matematika FMIPA-ITS.
5. Bapak Dr. Dieky Adzkiya, S.Si, M.Si, Bapak Drs. Kamiran, M.Si, Bapak Drs. Iis Herisman, M.Si sebagai dosen penguji Tugas Akhir yang telah memberikan kritik dan saran yang bersifat membangun dalam menyelesaikan Tugas Akhir ini.
6. Bapak Bagus yang telah membantu saya dan memperlancar dalam pencarian data kapal di PT. PAL INDONESIA.
7. Ibu, Mbak dan keluarga besar saya yang telah banyak mendukung dan memberi semangat saya dalam menjalani masa perkuliahan

8. Teman-teman mahasiswa frikha, daus, prima, ipan, mas fiyan, firdo, gery, lana, rozi, toembah, agus, romli, bayu, fatkhunur, sinar, wawan, uzu, gono, jojo, fadhlan , anshar, botak, aril, yoga, kodar, zani, oby, jon, jonbon dan seluruh mahasiswa Matematika 2013 serta keluarga STI-48 LAMBDA (yang tidak dapat saya sebutkan satu persatu) yang telah banyak mendukung baik dari doa maupun tindakan yang telah memberi semangat saya saat menjalani perkuliahan.
9. Semua pihak yang telah memberikan dukungan dan ilmu kepada penulis dalam penyelesaian Tugas Akhir ini.

Apabila dalam penulisan laporan ini ada kekurangan, kami mohon kritik dan saran demi penyempurnaan laporan kerja praktek di masa yang akan datang. Semoga laporan kerja praktek ini dapat bermanfaat bagi semua pihak. Amin.

Surabaya, Januari 2017

Penulis

DAFTAR ISI

	Hal
TUGAS AKHIR - SM141501	i
LEMBAR PENGESAHAN.....	v
Abstrak	vii
Abstract	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR SIMBOL.....	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	5
1.5 Manfaat.....	5
1.6 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	7
2.1 Penelitian Terdahulu.....	7
2.2 Kapal SSV BRP TARLAC (LD-601)	10
2.3 Model Matematika Kendali Haluan Kapal	11
2.4 Representasi Persamaan Keadaan pada Fungsi Transfer..	18
2.5 Teori kestabilan	22

	Hal
2.6 Keterkontrolan dan Keteramatan	23
2.7 Parameter Performansi Sistem Kendali	24
2.8 Kontrol PID	26
2.9 Kontrol Adaptif.....	28
2.10 <i>Self tuning regulator</i> (STR)	30
2.11 <i>Recursive Least Squares</i>	31
2.12 Metode Beda Hingga	34
BAB III METODE PENELITIAN	37
3.1 Studi Literatur	37
3.2 Persamaan Keadaan Model Matematika.....	37
3.3 Desain Kendali Haluan Kapal menggunakan PID adaptif	38
3.4 Simulasi dan Penarikan Kesimpulan.....	40
3.5 Penulisan Laporan Tugas Akhir.....	40
BAB IV ANALISIS DAN PEMBAHASAN	43
4.1 Persamaan Keadaan Kendali Haluan Kapal	43
4.1.1 Analisa Kestabilan Sudut <i>yaw</i>	45
4.1.2 Uji Keterkontrolan dan Keteramatan	46
4.2 Perancangan Desain Kendali PID Konvensional.....	47
4.3 Perancangan Desain Kendali PID adaptif.....	49
4.3.1 Sistem Diskrit Haluan Kapal SSV BRP TARLAC (LD-601)	50
4.3.2 Identifikasi Parameter Kendali PID	52

	Hal
4.3.3 <i>Online tuning</i> menggunakan RLS.....	57
4.4 Simulasi dan Analisis Hasil Perancangan Kendali	60
4.4.1 Simulasi dengan nilai <i>setpoint</i> berbeda	60
4.4.2 Simulasi dengan <i>Tracking setpoint</i>	71
4.4.3 Simulasi Dengan Gangguan	72
BAB V PENUTUP.....	75
5.1 Kesimpulan.....	75
5.2 Saran.....	76
DAFTAR PUSTAKA.....	77
LAMPIRAN A	81
LAMPIRAN B	83
LAMPIRAN C	84

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

	Hal
Gambar 2.1 Kapal SSV BRP TARLAC (LD-601)	10
Gambar 2.2 Enam Derajat Kebebasan Kapal	12
Gambar 2.3 Spesifikasi respon transien.....	25
Gambar 2.4 Diagram blok kontrol PID.....	26
Gambar 2.5 Sistem <i>direct adaptive</i>	29
Gambar 2.6 Sistem <i>indirect adaptive</i>	30
Gambar 2.7 Diagram blok kontrol adaptif.....	30
Gambar 2.8 Diagram blok <i>self tuning regulator</i>	31
Gambar 2.9 Jaringan titik metode beda hingga	34
Gambar 3.1 Diagram alir kendali PID adaptif	39
Gambar 3.2 Diagram alir metode penelitian.....	41
Gambar 4.1 <i>plant</i> sudut <i>heading</i> kapal	47
Gambar 4.2 Desain kendali PID	48
Gambar 4.3 Diagram blok PID Konvensional	48
Gambar 4.4 Desain Kendali PID konvensional	49
Gambar 4.5 <i>Plant</i> sistem diskrit	51
Gambar 4.6 Perancangan PID diskrit.....	55
Gambar 4.7 Kendali PID diskrit	56
Gambar 4.8 Diagram Blok RLS	57
Gambar 4.9 Desain <i>regressor</i>	59
Gambar 4.10 Desain PID adaptif.....	59
Gambar 4.11 Simulasi nilai awal 0 rad, <i>setpoint</i> 0,03 rad	61
Gambar 4.12 Sudut rudder percobaan 1	62
Gambar 4.13 Simulasi nilai awal 0,03 rad <i>setpoint</i> 0,09 rad	63
Gambar 4.14 Sudut <i>rudder</i> percobaan 2	64
Gambar 4.15 Simulasi nilai awal 0,09 rad, <i>setpoint</i> 0,18 rad	65
Gambar 4.16 Sudut <i>rudder</i> percobaan 3	66
Gambar 4.17 Simulasi nilai awal -0,03 rad, <i>setpoint</i> 0,03 rad...	67
Gambar 4.18 Sudut <i>rudder</i> percobaan 4	68

	Hal
Gambar 4.19 Simulasi nilai awal $-0,03$ rad, <i>setpoint</i> 0 rad	69
Gambar 4.20 Sudut <i>rudder</i> percobaan 5	70
Gambar 4.21 Uji Tracking kapal.....	72
Gambar 4.22 Respon kendali dengan gangguan	73

DAFTAR TABEL

	Hal
Tabel 2.1 Spesifikasi Kapal SSV BRP TARLAC (LD-601).....	11
Tabel 2.2 Variabel normalisasi pada sistem prime I	14
Tabel 2.3 Pengaruh <i>tuning</i> parameter PID	28
Tabel 4.1 Respon transien nilai awal 0 rad, <i>setpoint</i> 0,2 rad.....	62
Tabel 4.2 Perbandingan <i>settling time</i>	71

Halaman ini sengaja dikosongkan

DAFTAR SIMBOL

ψ	Sudut <i>yaw</i>
δ_R	Defleksi <i>rudder</i>
x	Variabel keadaan
y	Keluaran sistem
r	Kecepatan Sudut <i>yaw</i>
K_p	<i>gain proportional</i> pada pengendali PID
K_i	<i>gain integrator</i> pada pengendali PID
K_d	<i>gain derivative</i> pada pengendali PID
I_z	Momen Inersia pada sumbu <i>Z</i>
\mathcal{L}	Transformasi Laplace
\mathcal{L}^{-1}	Invers Transformasi Laplace
s	Variabel Laplace
T_i	<i>time integral</i>
T_d	<i>time derivatif</i>
C_B	Koefisien blok
x_g	<i>x coordinate of CG</i>
A_δ	Luas <i>rudder</i>
m	Massa kapal
u_0	Kecepatan jelajah kapal
$Y_{\dot{v}}$	Turunan gaya hidrodinamika <i>sway</i> terhadap \dot{v}
$Y_{\dot{r}}$	Turunan gaya hidrodinamika <i>sway</i> terhadap \dot{r}

$N_{\dot{r}}$	Turunan momen <i>yaw</i> terhadap \dot{r}
Y_v	Turunan gaya arah <i>sway</i> terhadap v
Y_r	Turunan gaya arah <i>yaw</i> terhadap r
N_v	Turunan momen <i>sway</i> terhadap v
$N_{\dot{v}}$	Turunan momen <i>sway</i> terhadap \dot{v}
N_r	Turunan momen <i>yaw</i> terhadap r
Y_{δ}	Turunan gaya <i>sway</i> terhadap sudut <i>rudder</i>
N_{δ}	Turunan momen <i>sway</i> terhadap sudut <i>rudder</i>
φ	Variabel regresi
θ	Parameter kendali PID
u	Input sistem
ρ	Massa jenis air laut
L	Panjang kapal
T	Kedalaman / tinggi kapal
B	Lebar kapal
t_r	<i>rise time</i>
t_s	<i>settling time</i>
t_d	<i>delay time</i>
M_p	<i>maximum overshoot</i>
t_p	<i>peak time</i>

BAB I

PENDAHULUAN

Bab ini membahas latar belakang yang mendasari penulisan Tugas Akhir. Didalamnya mencakup identifikasi permasalahan pada topik Tugas Akhir. Uraian ini bersifat umum yang menjelaskan secara ringkas hal-hal yang dilakukan pada penyelesaian Tugas Akhir. Informasi yang telah diperoleh tersebut kemudian dirumuskan menjadi permasalahan, kemudian diberikan asumsi-asumsi dan batasan-batasan untuk membatasi pembahasan pada Tugas Akhir ini.

1.1 Latar Belakang

Indonesia merupakan negara kepulauan terbesar didunia dengan luas wilayah laut Indonesia mencapai dua pertiga dari seluruh wilayah darat Indonesia dan memiliki garis pantai di hampir setiap pulau di Indonesia (\pm 81.000 km), serta Indonesia berada di daerah equator, antara dua benua Asia dan Australia, antara dua Samudera Pasifik dan Hindia, serta negara-negara Asia Tenggara. Secara geografis, Indonesia merupakan negara maritim, sesuai dengan UU No. 32 Tahun 2014 pasal 3 menegaskan Indonesia sebagai negara kepulauan berciri nusantara dan maritim. Salah satu potensi maritim Indonesia adalah industri maritim, menurut UU No. 32 Tahun 2014 pasal 27 ayat 3, industri maritim yang dimaksud dapat berupa galangan kapal, pengadaan dan pembuatan suku cadang, peralatan kapal, dan perawatan kapal. Salah satu peran industri maritim adalah pembuatan kapal perang dalam negeri maupun luar negeri [1]. Kapal perang SSV BRP TARLAC (LD-601) merupakan kapal perang buatan Indonesia yang di ekspor ke Filipina pada tanggal 13 Mei 2016 yang diserahkan kepada Kementerian Pertahanan Filipina. Kapal perang merupakan salah satu alutsista (alat

utama sistem pertahanan) yang digunakan untuk menjaga wilayah perairan negara. Mengingat wilayah laut rentan terhadap gangguan lingkungan seperti angin, gelombang, dan arus, saat beroperasi kapal harus mampu bermanuver untuk menjaga kestabilan ketika mendapatkan gangguan dari luar.

Saat melakukan manuver, gerak kapal mempunyai 6 derajat kebebasan atau *degrees of freedom* (DOF). Pergerakan ini berpusat pada tiga sumbu utama, antara lain: sumbu longitudinal (membujur), sumbu transversal (menyamping), dan sumbu vertikal (atas ke bawah). Secara umum gerakan yang dialami sebuah kapal ketika bermanuver ada dua macam, yaitu gerak translasi dan rotasi. Gerakan pada sumbu longitudinal adalah *surge* sebagai translasi dan *roll* sebagai rotasi. Sedangkan pada sumbu transversal adalah *sway* (translasi) dan *pitch* (rotasi). Dan *heave* dan *yaw* masing-masing sebagai translasi dan rotasi pada sumbu normal bumi [2]. Umumnya, sistem gerak kapal yang digunakan adalah tiga derajat kebebasan yaitu *surge*, *yaw*, dan *sway*. Dalam banyak kasus kecepatan *surge*, *sway*, dan *yaw* tak terkendali, dapat menghasilkan gerakan lain seperti *pitch*, *heave*, dan *roll* yang dapat menyebabkan guncangan keras dan kerusakan kargo pada kapal[3]. Oleh karena itu, dibutuhkan pengendali untuk meningkatkan tingkat keselamatan saat kapal melakukan manuver. Salah satu sistem kendali utama pada autopilot kapal adalah untuk mengendalikan haluan kapal atau biasa disebut *ship heading control*, sistem ini bertujuan untuk memimpin kapal pada lintasan yang diinginkan dan juga mampu mempersingkat 3-5% perjalanan dan mengurangi konsumsi bahan bakar [4].

Pada kendali haluan kapal, variabel yang dikendalikan hanya dalam satu derajat kebebasan (DOF), yaitu kecepatan *yaw* dengan asumsi gerak *sway*, *surge*, *heave*,

roll, *pitch* tidak berpengaruh pada manuver kapal. Pendekatan fungsi transfer yang sering digunakan untuk kendali haluan kapal adalah model Nomoto dengan mempertimbangkan gerak *yaw* dan satu *input* kontrol[2]. Salah satu pengendali yang sering digunakan untuk sistem autopilot adalah pengendali PID. Sistem kendali PID autopilot pertama kali disajikan oleh Minorsky dan Sperry pada tahun 1922 untuk pada model *single-input single-output* (SISO), di mana *heading* (sudut *yaw*) pada kapal diukur dengan *gyrocompass*. Autopilot membandingkan *setpoint* (*heading* yang diinginkan) dengan *heading* dari pengukuran, dan *error* digunakan sebagai *input* ke *controller*. *Output* dari *controller* kemudian ditransmisikan ke *rudder servo* yang menghasilkan sinyal kontrol yang sesuai untuk menggerakkan kemudi kapal [5]. Kontrol PID terdiri dari tiga bagian yaitu Proporsional, Integral dan Derivatif. PID merupakan salah satu *controller* dan *stabilizer* yang sering digunakan di industri yang sederhana dan mudah untuk diimplementasikan, namun dalam beberapa kasus parameter sistem atau kondisi operasi yang tidak menentu, atau bervariasi selama beroperasi, kontrol PID konvensional tidak akan mengubah gain untuk mengatasi perubahan sistem[6]. Oleh karena itu, perlu dikembangkan kontroler PID konvensional dan desain kontrol baru yang bisa memperkirakan beberapa ketidakpastian dalam sistem kemudian mengubah kontrol secara otomatis, atau biasa disebut sistem kendali adaptif [7]. Dalam beberapa literatur, teknik penyetelan kontrol PID dibagi menjadi dua kelompok yakni metode secara *off-line* Zeigler-Nichols dan secara *online* atau biasa disebut kontrol PID adaptif. Kontrol dengan menggunakan PID adaptif dapat menyesuaikan gain PID untuk menyesuaikan sistem agar mengikuti kinerja yang diinginkan bahkan dengan adanya beberapa perubahan dalam karakteristik sistem[6].

Pada Tugas Akhir ini, akan dilakukan perancangan desain kontrol PID adaptif dengan menggunakan kontrol adaptif *self tuning regulator* untuk menghasilkan sinyal kontrol pada permasalahan kendali haluan kapal, selanjutnya estimator parameter pengendali PID menggunakan algoritma *recursive least squares*. Hasil yang diharapkan dari metode ini adalah menunjukan kinerja dari pengendali PID adaptif untuk mengendalikan haluan kapal pada saat kapal mengalami gangguan pada sistem, selanjutnya kendali PID adaptif akan dibandingkan dengan PID konvensional untuk membandingkan kinerja dari masing-masing pengendali. Kinerja pengendali diukur dari karakteristik *respon transient* masing-masing pengendali terhadap *setpoint* yang diberikan. Selanjutnya metode yang diusulkan akan diterapkan pada model kendali haluan kapal perang SSV BRP TARLAC (LD-601).

1.2 Rumusan Masalah

Rumusan masalah yang dibahas dalam Tugas Akhir ini, yaitu:

1. Bagaimana penerapan *adaptive PID controller* pada kendali haluan kapal (*ship heading control*).
2. Bagaimana hasil simulasi dari penerapan *adaptive PID controller* pada kendali haluan kapal (*ship heading control*) dengan menggunakan *software* MATLAB.

1.3 Batasan Masalah

Batasan-batasan masalah yang digunakan dalam Tugas Akhir ini, yaitu:

1. Model matematika dengan menggunakan pendekatan dari model Nomoto orde 2.
2. Kapal yang digunakan adalah Kapal Perang SSV BRP TARLAC (LD-601).

1.4 Tujuan

Tujuan yang ingin dicapai dalam Tugas Akhir, yaitu:

1. Menerapkan desain *adaptive PID controller* pada kendali haluan kapal (*ship heading controller*).
2. Mengetahui hasil simulasi dan penerapan desain *adaptive PID controller* pada kendali haluan kapal (*ship heading control*).

1.5 Manfaat

Manfaat yang diharapkan dari Tugas Akhir ini, yaitu:

1. Diperoleh pengetahuan untuk menerapkan desain *adaptive PID controller* pada kasus kendali haluan kapal (*ship heading control*).
2. Sebagai rujukan untuk sistem navigasi, panduan dan kendali pada haluan kapal.

1.6 Sistematika Penulisan

Penulisan Tugas Akhir ini disusun dalam lima bab, yaitu:

1. BAB I PENDAHULUAN

Bab ini berisi tentang gambaran umum dari penulisan Tugas Akhir yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Dalam bab ini diuraikan model kendali haluan kapal menggunakan pendekatan fungsi transfer model Nomoto orde 2 pada kapal SSV BRP TARLAC (LD-601), Kendali PID, Kendali adaptif, metode *recursive Least Squares*.

3. BAB III METODE PENELITIAN

Dalam bab ini dijelaskan tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir, tahapan-tahapan tersebut antara lain, pemodelan matematika persamaan

keadaan kapal SSV BRP TARLAC (LD-601), perancangan pengendali PID, proses *online tuning* parameter PID, perancangan desain sistem kendali PID adaptif. Selanjutnya dilakukan simulasi dan analisis terhadap hasil yang diperoleh. Tahap terakhir adalah melakukan penarikan kesimpulan dari hasil analisis dan pembahasan yang telah dilakukan.

4. BAB IV ANALISIS DAN PEMBAHASAN

Dalam bab ini berisi tentang persamaan keadaan yang dibentuk berdasarkan fungsi transfer model Nomoto orde 2 dengan sudut *yaw* sebagai keluaran sistem, uji keterkontrolan (*controllability*) sistem, uji keteramatan (*observability*) sistem, perancangan dan simulasi sistem kendali PID adaptif pada kapal SSV BRP TARLAC (LD-601) serta penjelasan mengenai hasil simulasi.

5. BAB V PENUTUP

Dalam bab ini berisi kesimpulan akhir yang diperoleh dari Tugas Akhir serta saran untuk pengembangan penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai dasar teori yang digunakan dalam penyusunan Tugas Akhir ini. Dasar teori yang digunakan meliputi dalam Penelitian terdahulu, spesifikasi kapal SSV BRP TARLAC (LD-601), fungsi transfer Nomoto orde 2, Kendali PID, Kendali PID adaptif proses *online tuning* menggunakan *recursive least squares*, Metode beda hingga.

2.1 Penelitian Terdahulu

Tinjauan pustaka yang digunakan dalam tugas akhir ini adalah beberapa penelitian yang relevan dengan tema yang diambil, Pada penelitian sebelumnya, telah diperoleh model matematika haluan kapal oleh M. Ardi Firmansyah tahun 2016 untuk kapal perang SSV BRP TARLAC (LD-601) yang disimulasikan menggunakan *software* MATLAB. Model matematika diperoleh dari penurunan rumus model linear kemudi kapal Davidson dan Schiff dan pendekatan fungsi transfer model Nomoto orde 2, diperoleh fungsi transfer yang selanjutnya diubah dalam bentuk *state-space* dengan variabel yang diamati adalah kecepatan *yaw* dan sudut *yaw*. Dari hasil simulasi diperoleh bahwa model matematika haluan kapal merupakan sistem yang stabil, namun dalam penelitian ini belum terdapat perancangan kendali haluan kapal perang SSV BRP TARLAC (LD-601), oleh karena itu perlu dirancang sebuah pengendali yang mampu mengendalikan sistem[14].

Autopilot pada kendali manuver kapal telah berkembang sejak 1922 oleh Minorsky dan Sperry dengan menggunakan kontrol PID sebagai kontroler, Perkembangan selanjutnya adalah *linear steering* yang diturunkan oleh Davidson dan Schiff (1946), Nomoto

(1957), dan juga *nonlinear steering* oleh Abkowitz (1964), Norrbm (1970). Perkembangan dari autopilot kapal mampu memberikan respon yang baik terhadap kontrol pada kapal, beberapa tahun terakhir, muncul teori-teori sistem kontrol baru yang mampu membantu kinerja dari sistem autopilot pada kapal diantaranya kendali LQG dan H_∞ [2].

Kontrol PID sendiri merupakan kontrol yang paling sering dan yang paling praktis digunakan di industri, dalam aplikasi kontrol proses, lebih dari 95% menggunakan pengendali PID (Astrom dan Hagglund, 1995). PID dalam implementasinya telah direkomendasikan untuk kontrol proses dengan orde rendah sampai menengah, dengan penundaan waktu kecil. Kontrol PID merupakan pengendali yang mampu mempercepat respon *transient*, menghilangkan *error steady state*, memberikan efek redaman, dan mengurangi *offset*[10].

Namun beberapa tahun terakhir, telah banyak penelitian bahwa control PID konvensional tidak mampu memperbaiki parameter *gain* pengendali secara kontinu pada saat perubahan sistem pada *plant*, pada tahun 2014 telah dilakukan penelitian tentang kontrol adaptif PID menggunakan *recursive least squares* pada sistem stabil dan tidak stabil pada kontrol SISO (*single input single output*) oleh Rania A. Fahmy et.al. Pada penelitian tersebut RLS digunakan untuk memperbaiki gain PID untuk memaksa menuju sistem aktual. RLS merupakan algoritma untuk estimasi parameter *online* yang sering digunakan karena mempunyai kecepatan untuk menuju nilai yang konvergen. Dari hasil simulasi menunjukkan PID adaptif menggunakan RLS mampu bekerja dengan baik pada sistem stabil dan tidak stabil, dari hasil simulasi juga menunjukkan efisiensi sistem ketika perubahan parameter sistem dengan menggunakan RLS [6].

Pada tahun 2009, Xiaoyu Liu et.al melakukan penelitian tentang desain pada PID adaptif menggunakan

metode *least squares*. Pada penelitian ini, PID adaptif diterapkan pada model dengan sistem berorde 2 yang disimulasikan menggunakan MATLAB. Dari hasil simulasi menunjukkan bahwa PID adaptif menggunakan estimasi *least squares* memberikan respon yang cepat, *over regulation* dan robust, bahkan pengendali mampu menyesuaikan secara cepat berdasarkan perubahan pada sistem[7]. Setelah itu pada tahun 2012, telah dilakukan penelitian oleh, Yuji wakasa et.al tentang pengendali dengan *tuning* secara *online* menggunakan FRIT dan *recursive least squares*. Pada penelitian ini, RLS menggunakan faktor *forgetting* untuk memberikan bobot yang lebih pada data yang baru. Pada penelitian ini, penulis membandingkan antara FRIT standar dan algoritma RLS, dari hasil simulasi menunjukkan, algoritma yang diusulkan dapat dilakukan secara terus menerus tanpa berhenti saat kontrol beroperasi atau tanpa mengubah parameter kontroler tiba-tiba, yang berarti bahwa metode yang diusulkan lebih praktis daripada konvensional[11].

Pada tahun 1983, telah dilakukan penelitian tentang desain autopilot untuk kendali kapal oleh Lim dan Forsythe dengan menggunakan desain kontrol menggunakan *self-tuning* yang diterapkan pada autopilot kapal untuk beberapa permasalahan, diantaranya untuk meminimalkan *error* pada *heading*. Parameter pada sistem diidentifikasi menggunakan *recursive least squares* (RLS) untuk meningkatkan sifat adaptif pada autopilot, dari hasil simulasi menunjukkan bahwa metode *self-tuning* mampu memenuhi kebutuhan kemudi untuk autopilot model kapal cargo. Namun masih perlu dikembangkan metode *self-tuning* autopilot yang dapat mengidentifikasi sistem secara *online* dan mampu beradaptasi terhadap perubahan yang terjadi pada sistem dan gangguan luar [8].

Kendali PID adaptif *self tuning regulator* telah diimplementasikan pada pengaturan kecepatan motor induksi tiga fasa oleh Halim Mudia pada tahun 2012[12], pada penelitian ini kendali PID adaptif mampu mencapai *setpoint* dengan cepat dan parameter PID mampu menyesuaikan pada saat sistem mengalami gangguan, selain itu pada tahun 2014 PID adaptif diimplementasikan untuk mengendalikan frekuensi pembangkit listrik mini hidro oleh Nurul Aini [12], hasil penelitian menunjukkan kinerja PID adaptif sangat baik dan mampu diimplementasikan pada pembangkit listrik mini hidro.

2.2 Kapal SSV BRP TARLAC (LD-601)

Kapal Perang *Strategic Sealift Vessel* ‘BRP TARLAC (LD-601)’ merupakan kapal perang buatan Indonesia yang diekspor ke Filipina pada tanggal 13 Mei 2016 yang diserahkan kepada Kementerian Pertahanan Filipina merupakan pengembangan dari kapal pengangkut jenis *Landing Platform Dock* (LPD) yang didesain dengan panjang 122,970 meter dan lebar 21,796 meter [14]. Dengan gambar dan spesifikasi kapal dapat dilihat pada gambar 2.1 dan tabel 2.1.



Gambar 2.1 Kapal SSV BRP TARLAC (LD-601) [14]

Berikut akan ditampilkan data spesifikasi Kapal SSV BRP TARLAC (LD-601) [14].

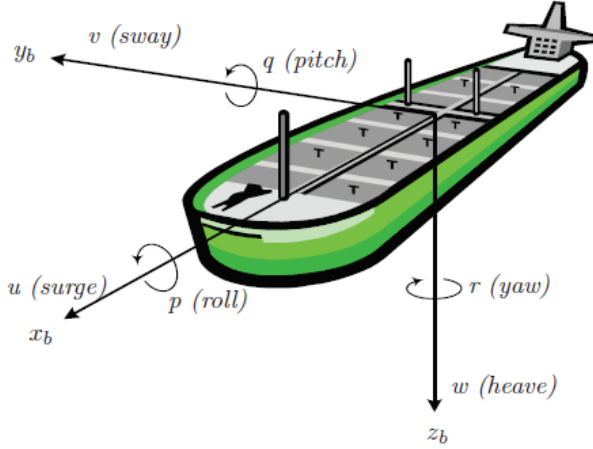
Tabel 2.1 Spesifikasi Kapal SSV BRP TARLAC (LD-601)

DATA KAPAL SSV BRP TARLAC (LD-601)	
<i>length of all (LOA)</i>	122,970 m
<i>length between perpendicular (LBP)</i>	107, 424 m
<i>breadth (lebar)</i>	21,796 m
<i>depth (T)</i>	11, 298 m
<i>Draught</i>	5,00 m
<i>max speed</i>	16 knot
<i>diameter propeller</i>	3 m
<i>sudut rudder</i>	max 35°
<i>rudder area (A_δ)</i>	5,5 m ²
<i>cruising speed</i>	13 knot
<i>light ship condition displacement</i>	4676,4 ton
<i>koefisien blok (C_B)</i>	0,6
<i>x coordinate of CG (x_G)</i>	5,15 m
<i>moment inertia in yaw (I_z)</i>	3,54x10 ⁹ kg m ²
<i>full load displacement</i>	9271,5 ton
<i>volume of light ship displacement</i>	4562, 34 m ³

2.3 Model Matematika Kendali Haluan Kapal

Secara umum gerakan kapal dibagi menjadi 6 DOF (*degrees of freedom*) dimana dari 6 DOF dibagi lagi menjadi dua macam gerakan yaitu gerak translasi dan rotasi. Gerak translasi dibagi menjadi tiga yaitu *surge*, *sway* dan *heave*. Sedangkan gerak rotasi dibagi menjadi tiga yaitu *roll*, *pitch*, dan *yaw* , namun pada kenyataanya kapal yang berada pada permukaan hanya mempertimbangkan 3 derajat kebebasan *surge*, *sway*, dan *yaw*[2].

6 DOF atau derajat kebebasan pada saat kapal melakukan manuver di laut ditunjukkan pada gambar 2.2



Gambar 2.2 Enam Derajat Kebebasan Kapal [2]

Bentuk umum persamaan maneuvering kapal dinyatakan dalam bentuk:

$$M \dot{v} + N(u_0)v = b\delta_R \quad (2.1)$$

dengan matriks $N(u_0)$ adalah penjumlahan gaya redaman linier dan gaya Coriolis dan sentripetal $C(u_0)$ (dengan syarat penambahan mu_0 dan $mx_g u_0$)

$$N(u_0) = C(u_0) + D \quad (2.2)$$

$v = [v, r]^T$ adalah vektor keadaan, δ_R adalah sudut rudder dan:

$$M = \begin{bmatrix} m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ mx_g - N_{\dot{v}} & I_z - N_{\dot{r}} \end{bmatrix} \quad (2.3)$$

$$N(u_0) = \begin{bmatrix} -Y_v & mu_0 - Y_r \\ -N_v & mx_g u_0 - N_r \end{bmatrix} \quad (2.4)$$

$$b = \begin{bmatrix} Y_{\delta} \\ N_{\delta} \end{bmatrix} \quad (2.5)$$

Selanjutnya dengan menggunakan pendekatan *slender body strip* turunan koefisien hidrodinamika yang telah dikembangkan oleh Smitt (1970), Norrbinn (1971) dan Inoue (1981) dan dikemukakan oleh Clarke (1982) ditunjukkan dalam bentuk persamaan (2.6-2.15). Koefisien hidrodinamika pada persamaan (2.6-2.15) merupakan bentuk non dimensional yang diturunkan dengan sistem prime I. Untuk memperoleh besaran gaya dikalikan dengan $\frac{1}{2}\rho U^2 L^2$ momen dikalikan dengan $\frac{1}{2}\rho U^2 L^3$, besaran massa dikalikan dengan $\frac{1}{2}\rho L^3$, besaran momen inersia dikalikan dengan $\frac{1}{2}\rho L^5$. Dimana ρ = massa jenis air laut (1025 kg/m^3), L = *Lenght between perpendicular* = panjang kapal, U = kecepatan servis kapal, B = lebar kapal, T = kedalaman/tinggi kapal, C_B = koefisien blok. Sehingga diperoleh:

$$\frac{-Y_v'}{\pi (T/L)^2} = 1 + 0,16 \frac{C_B B}{T} - 5,1 \left(\frac{B}{L}\right)^2 \quad (2.6)$$

$$\frac{-Y_r'}{\pi (T/L)^2} = 0,67 \left(\frac{B}{L}\right) - 0,0033 \left(\frac{B}{T}\right)^2 \quad (2.7)$$

$$\frac{-N_v'}{\pi (T/L)^2} = 1,1 \left(\frac{B}{L}\right) - 0,041 \left(\frac{B}{T}\right) \quad (2.8)$$

$$\frac{-N_r'}{\pi (T/L)^2} = \frac{1}{12} + 0,017 \frac{C_B B}{T} - 0,33 \left(\frac{B}{L}\right) \quad (2.9)$$

$$\frac{-Y_v'}{\pi (T/L)^2} = 1 + 0,4 \frac{C_B B}{T} \quad (2.10)$$

$$\frac{-Y_r'}{\pi (T/L)^2} = -\frac{1}{2} + 2,2 \left(\frac{B}{L}\right) - 0,08 \left(\frac{B}{T}\right) \quad (2.11)$$

$$\frac{-N_v'}{\pi (T/L)^2} = \frac{1}{2} + 2,4 \left(\frac{T}{L}\right) \quad (2.12)$$

$$\frac{-N_r'}{\pi (T/L)^2} = \frac{1}{4} + 0,039 \frac{B}{T} - 0,56 \left(\frac{B}{L}\right) \quad (2.13)$$

sedangkan untuk gaya dan momen yang diakibatkan *rudder* adalah:

$$Y_{\delta}' = \rho \frac{\pi A_{\delta}}{4 L T} \quad (2.14)$$

$$N_{\delta}' = -\frac{1}{2} Y_{\delta}' \quad (2.15)$$

Berikut akan ditunjukkan variabel yang digunakan untuk normalisasi dengan menggunakan sistem prime I dengan satuan yang telah ditentukan:

Satuan	Sistem Prime I
Panjang	L
Massa	$\frac{\rho}{2} L^3$
Momen Inersia	$\frac{\rho}{2} L^5$
Waktu	$\frac{L}{U}$
Sudut	1
Kecepatan linier	U
Kecepatan sudut	$\frac{U}{L}$
Percepatan linier	$\frac{U^2}{L}$
Percepatan sudut	$\frac{U^2}{L^2}$
Gaya	$\frac{\rho}{2} U^2 L^2$
Momen	$\frac{\rho}{2} U^2 L^3$
Posisi	L

Tabel 2.2 Variabel normalisasi pada sistem prime I bentuk normalisasi pada model persamaan (2.1) sesuai dengan sistem prime I diperoleh:

$$M' \dot{v}' + N'(u_0') v' = b' \delta_R' \quad (2.16)$$

notasi (') merupakan variabel nondimensional yang telah melalui prose normalisasi, dengan $v' = [v', r']^T$ dan

$$\mathbf{M}' = \begin{bmatrix} m' - Y_v' & m'x_g' - Y_r' \\ m'x_g' - N_v' & I_z' - N_r' \end{bmatrix} \quad (2.17)$$

$$\mathbf{N}'(u_0') = \begin{bmatrix} -Y_v' & m'u_0' - Y_r' \\ -N_v' & m'x_g'u_0' - N_r' \end{bmatrix} \quad (2.18)$$

$$\mathbf{b}' = \begin{bmatrix} Y_\delta' \\ N_\delta' \end{bmatrix} \quad (2.19)$$

dengan,

$$u_0' = \frac{u_0}{U} = \frac{u_0}{\sqrt{(u_0 + \Delta u)^2 + \Delta v^2}} \approx 1 \quad (2.20)$$

untuk nilai Δu dan Δv kecil, sistem nondimensional (2.16) dapat dihubungkan dengan persamaan (2.1) dengan menerapkan transformasi yang diperoleh dengan menggunakan sistem prime I, sehingga diperoleh variabel non dimensional sebagai berikut:

$$v = U v' \quad (2.21)$$

$$r = \frac{U}{L} r' \quad (2.22)$$

$$\delta_R = \delta_R' \quad (2.23)$$

$$x_G = x_G' L \quad (2.24)$$

$$m = m' \frac{\rho}{2} L^3 \quad (2.25)$$

$$I_z = I_z' \frac{\rho}{2} \quad (2.26)$$

Dalam desain kendali haluan kapal (*ship heading control*), model Nomoto merupakan model yang paling umum digunakan. Model ini hanya mempertimbangkan satu derajat kebebasan (DOF) sistem dinamik kapal, yakni r (kecepatan *yaw*), dan satu *input* kontrol, yakni δ_R (sudut kemudi).

Pada Tugas Akhir ini, model matematika dari kendali haluan kapal didapatkan dari pendekatan fungsi transfer dari model Nomoto, Taguchi, Honda dan Hirano (1957) orde dua. Model ini mengeliminasi kecepatan *sway* v dari persamaan linear dinamika kapal untuk menghasilkan fungsi transfer Nomoto orde ke dua antara r dan δ_R [5].

Fungsi transfer model Nomoto orde 2 ditunjukkan pada persamaan berikut:

$$\frac{\psi(s)}{\delta_R(s)} = \frac{K_R(1 + T_3s)}{s(1 + T_1s)(1 + T_2s)} \quad (2.27)$$

karena $r(s) = s \psi(s)$, persamaan (2.27) dapat ditulis menjadi:

$$\frac{r(s)}{\delta_R(s)} = \frac{K_R(1 + T_3s)}{(1 + T_1s)(1 + T_2s)} \quad (2.28)$$

persamaan (2.28) dapat ditulis menjadi:

$$\frac{r(s)}{\delta_R(s)} = \frac{K_R T_3 s + K_R}{T_1 T_2 s^2 + (T_1 + T_2)s + 1} \quad (2.29)$$

dengan,

$$T_1 T_2 = \frac{\det(\mathbf{M}')}{\det(\mathbf{N}')} \quad (2.30)$$

$$\begin{aligned} & T_1 + T_2 \\ &= \frac{n_{11}'m_{22}' + n_{22}'m_{11}' - n_{12}'m_{21}' - n_{21}'m_{12}'}{\det(\mathbf{N}')} \end{aligned} \quad (2.31)$$

$$K_R = \frac{n_{21}'b_1' - n_{11}'b_2'}{\det(\mathbf{N}')} \quad (2.32)$$

$$K_RT_3 = \frac{m_{21}'b_1' - m_{11}'b_2'}{\det(\mathbf{N}')} \quad (2.33)$$

untuk elemen $m_{i,j}'$, $n_{i,j}'$ dan b_i' untuk ($i = 1,2$ dan $j = 1,2$) didapatkan dari elemen baris dan kolom pada matriks persamaan (2.17 sampai 2.19), elemen i merupakan representasi dari baris matriks dan elemen j merupakan representasi dari kolom matriks, dengan determinan dari matriks \mathbf{M}' dan \mathbf{N}' diberikan pada persamaan (2.34) dan (2.35)

$$\det(\mathbf{N}') = Y_v'(N_r' - m'x_g'u_0') - N_v'(Y_r' - m'u_0') \quad (2.34)$$

$$\det(\mathbf{M}') = (m' - Y_v')(I_z' - N_r') - (m'x_g' - N_v')(m'x_g' - Y_r') \quad (2.35)$$

Dengan menggunakan data kapal SSV BRP TARLAC (LD-601) pada tabel 2.1 dan dengan menggunakan pendekatan model Nomoto, fungsi transfer kendali haluan kapal SSV BRP TARLAC (LD-601) ditunjukkan pada persamaan berikut[14]:

$$\frac{r(s)}{\delta_R(s)} = \frac{244,2.s + 484,6}{0,1203.s^2 + 0,7312.s + 1} \quad (2.36)$$

dengan $r(s) = \dot{\psi}(s)$ merupakan kecepatan yaw dan $\delta_R(s)$ merupakan defleksi *rudder* pada kapal sebagai *input*

sistem. Selanjutnya dari persamaan (2.36) sistem diubah ke persamaan ruang keadaan

$$\dot{x} = \mathbf{A}x + \mathbf{B}u \quad (2.37)$$

dengan \mathbf{A} adalah matriks *state* ordo 2×2 , \mathbf{B} matriks *input* ordo 2×1 dan $x = \begin{bmatrix} \psi \\ r \end{bmatrix}$; $r = \dot{\psi}$.

2.4 Representasi Persamaan Keadaan pada Fungsi Transfer

Dalam teori kontrol, fungsi transfer digunakan untuk menunjukkan hubungan *input-output* dari komponen atau sistem yang dapat dijelaskan secara linear, waktu-invariant, persamaan diferensial. Fungsi transfer didefinisikan sebagai rasio dari Transformasi Laplace dari *output* (fungsi respon) terhadap Transformasi Laplace dari *input* (kendali fungsi) dengan asumsi bahwa semua kondisi awal adalah nol.

Dengan mempertimbangkan sistem *time-invariant* didefinisikan persamaan diferensial sebagai berikut:

$$\begin{aligned} a_0 y^n + a_1 y^{n-1} + \dots + a_{n-1} \dot{y} + a_n y \\ = b_0 x^m + b_1 x^{m-1} + \dots + b_{m-1} \dot{x} \\ + b_m x \quad (n \geq m) \end{aligned} \quad (2.38)$$

dengan y adalah keluaran dari sistem dan x adalah masukan. Fungsi transfer pada sistem adalah perbandingan dari transformasi Laplace keluaran dan transformasi Laplace masukan, dengan semua kondisi awal sama dengan nol [18].

$$G(s) = \frac{\mathcal{L}[\text{output}]}{\mathcal{L}[\text{input}]} | \text{kondisi awal} = 0$$

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^m + b_1 s^{m-1} + b_{m-1} s + b_m}{a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n} \quad (2.39)$$

untuk kasus $m < n$

dengan,

$Y(S)$ = isyarat keluaran

$U(S)$ = isyarat masukan

Selanjutnya dengan memberikan isyarat sembarang $x(t) = \mathcal{L}^{-1}X(s)$, dengan \mathcal{L}^{-1} adalah invers transformasi laplace maka persamaan (2.39) menjadi

$$\frac{Y(S)}{U(S)} = \frac{(b_0S^m + b_1S^{m-1} + \dots + b_{m-1}S + b_m)X(S)}{(a_0S^n + a_1S^{n-1} + \dots + a_{n-1}S + a_n)X(S)} \quad (2.40)$$

maka diperoleh persamaan:

$$U(s) = (a_0S^n + a_1S^{n-1} + \dots + a_{n-1}S + a_n)X(S) \quad (2.41)$$

dengan asumsi keadaan awal $x(0) = \text{nol}$, maka *inverse* transformasi Laplace akan menghasilkan

$$\begin{aligned} U(t) = & a_0 \frac{d^n x(t)}{dt^n} + a_1 \frac{d^{n-1} x(t)}{dt^{n-1}} + a_2 \frac{d^{n-2} x(t)}{dt^{n-2}} + \dots \\ & + a_{n-1} \frac{dx(t)}{dt} + a_n x(t) \end{aligned} \quad (2.42)$$

untuk memenuhi bentuk umum model ruang keadaan, maka diasumsikan

$$U(t) \triangleq u$$

$$x(t) \triangleq x_1 \rightarrow \dot{x}_1 = \frac{dx_1(t)}{dt} = \frac{dx(t)}{dt} \triangleq x_2$$

$$\frac{dx(t)}{dt} \triangleq x_2 \rightarrow \dot{x}_2 = \frac{dx_2}{dt} = \frac{d^2 x(t)}{dt^2} \triangleq x_3$$

$$\frac{d^2 x(t)}{dt} \triangleq x_3 \rightarrow \dot{x}_3 = \frac{dx_3}{dt} = \frac{d^3 x(t)}{dt^3} \triangleq x_4$$

$$\vdots$$

$$\frac{d^{n-2}x(t)}{dt^{n-2}} \triangleq x_{n-1} \rightarrow \dot{x}_{n-1} = \frac{dx_{n-1}}{dt} = \frac{d^{n-1}x(t)}{dt^{n-1}} \triangleq x_n$$

$$\frac{d^{n-1}x(t)}{dt^{n-1}} \triangleq x_n \rightarrow \dot{x}_n = \frac{dx_n(t)}{dt} = \frac{d^n x(t)}{dt^n}$$

dengan,

$$\begin{aligned} \frac{d^n x(t)}{dt^n} = & -\frac{a_1}{a_0} \frac{d^{n-1}x(t)}{dt^{n-1}} - \frac{a_2}{a_0} \frac{d^{n-2}x(t)}{dt^{n-2}} - \dots \\ & - \frac{a_{n-1}}{a_0} \frac{dx(t)}{dt} - \frac{a_n}{a_0} x(t) + \frac{1}{a_0} u \end{aligned}$$

sehingga diperoleh

$$\dot{x}_n = -\frac{a_n}{a_0} x_1 - \frac{a_{n-1}}{a_0} x_2 - \dots - \frac{a_2}{a_0} x_{n-1} - \frac{a_1}{a_0} x_n + \frac{1}{a_0} u$$

sehingga diperoleh Persamaan keadaan $\dot{x} = Ax + Bu$ pada persamaan berikut dengan

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} \quad (2.43)$$

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -\frac{a_n}{a_0} & -\frac{a_{n-1}}{a_0} & -\frac{a_{n-2}}{a_0} & \dots & -\frac{a_2}{a_0} & -\frac{a_1}{a_0} \end{bmatrix} \quad (2.44)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \frac{1}{a_0} \end{bmatrix} \quad (2.45)$$

Contoh 3.1

Dari fungsi transfer akan diubah ke dalam bentuk ruang keadaan

$$G(S) = \frac{2S + 5}{3S^2 + 4S + 1}$$

Persamaan (2.40) ekuivalen dengan $G(s)$ maka dapat diperoleh $m = 1; n = 2$ untuk fungsi transfer diatas dan juga diperoleh parameter-parameter sebagai berikut

$$b_0 = 2; b_1 = 5$$

$$a_0 = 3; a_1 = 4; a_2 = 1$$

karena nilai $n = 2$ pada fungsi transfer diatas, dengan melihat kembali persamaan (2.43) diperoleh

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}$$

selanjutnya pandang persamaan (2.44) dan (2.45), dengan menggunakan parameter pada persamaan fungsi transfer diatas diperoleh matriks A dan B sebagai berikut:

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{1}{3} & \frac{4}{3} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1 \\ \frac{1}{3} \end{bmatrix}$$

sehingga diperoleh persamaan ruang keadaan dari contoh fungsi transfer adalah sebagai berikut:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{3} & \frac{4}{3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ \frac{1}{3} \end{bmatrix} u$$

selanjutnya, untuk memperoleh pengamatan dari keluaran sistem dari x_1 saja dan mengabaikan x_2 maka dapat dipilih matriks C sebagai berikut:

$$C = [1 \quad 0]$$

sehingga diperoleh persamaan dari keluaran sistem

$$y = Cx + D$$

$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]$$

dengan ordo matriks

$$A_{n \times n} ; B_{n \times m} ; C_{k \times n} ; D_{k \times m}$$

2.5 Teori kestabilan

Sebuah sistem dikatakan stabil jika sistem tersebut akan tetap dalam keadaan diam atau berhenti kecuali jika dirangsang atau dengan kata lain sistem dikatakan stabil jika *natural response* mendekati nol, beberapa metode telah diusulkan untuk menentukan kestabilan sistem, diantaranya dengan melihat nilai karakteristik sistem atau nilai eigen, berikut merupakan teorema kestabilan yang digunakan pada Tugas Akhir ini. Diberikan suatu sistem persamaan linear *time invariant* yang berdimensi- n yang disajikan oleh persamaan:

$$\begin{cases} \dot{x} = Ax(t) + Bu(t) \\ y = Cx(t) + Du(t) \end{cases} \quad (2.46)$$

Maka dapat ditentukan kestabilan suatu sistem tersebut melalui nilai karakteristik (λ_i). Dengan nilai karakteristik dapat diperoleh melalui rumus:

$$\det(\lambda I - A) = 0$$

dengan: λ = nilai karakteristik

I = matriks identitas

A = matriks ber ordo $n \times n$ bernilai real

Teorema yang berkaitan dengan kestabilan untuk sistem linear *time-invariant* adalah [18]:

Teorema 2.5.1 Diberikan persamaan diferensial $\dot{x} = Ax$, dengan matriks A berukuran $n \times n$ dan mempunyai nilai karakteristik yang berbeda $\lambda_1, \dots, \lambda_k$ ($k \leq n$).

- Titik asal $\bar{x} = 0$ adalah stabil asimtotik bila dan hanya bila $\text{Re}(\lambda_i) < 0$ untuk semua $i = 1, \dots, k$.
- Titik asal $\bar{x} = 0$ adalah stabil bila dan hanya bila $\text{Re}(\lambda_i) \leq 0$ untuk semua $i = 1, \dots, k$ dan untuk semua λ_i dengan $\text{Re}(\lambda_i) = 0$ multiplisitas aljabar sama dengan multiplisitas geometrinya.
- Titik asal $\bar{x} = 0$ adalah tak stabil bila dan hanya bila $\text{Re}(\lambda_i) > 0$ untuk beberapa $i = 1, \dots, k$ atau ada λ_i dengan $\text{Re}(\lambda_i) = 0$ dan multiplisitas aljabar lebih besar dari multiplisitas geometrinya.

2.6 Keterkontrolan dan Keteramatan

Diberikan sistem linear *time-invariant* yang berdimensi- n yang disajikan pada persamaan (2.46)

Definisi 2.5.1. Sistem linear (2.46) dikatakan terkontrol bila untuk setiap keadaan sebarang $x(0) = x_0$ ada masukan $u(t)$ yang tidak dibatasi mentransfer keadaan x_0 ke sebarang keadaan akhir $x(t_1) = x_1$ dengan waktu akhir t_1 hingga [18].

Syarat cukup dan perlu suatu sistem linear (2.46) dikatakan terkontrol, jika matriks:

$$M_c = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B]$$

mempunyai *rank* sama dengan n .

Definisi 2.5.2. Bila setiap keadaan awal $\mathbf{x}(0) = \mathbf{x}_0$ secara tunggal dapat diamati dari setiap pengukuran keluaran sistem (2/46) dari waktu $t = 0$ ke $t = t_1$, maka sistem dikatakan “teramati”[18].

Syarat cukup dan perlu suatu sistem linear (2.46) dikatakan teramati, jika matriks:

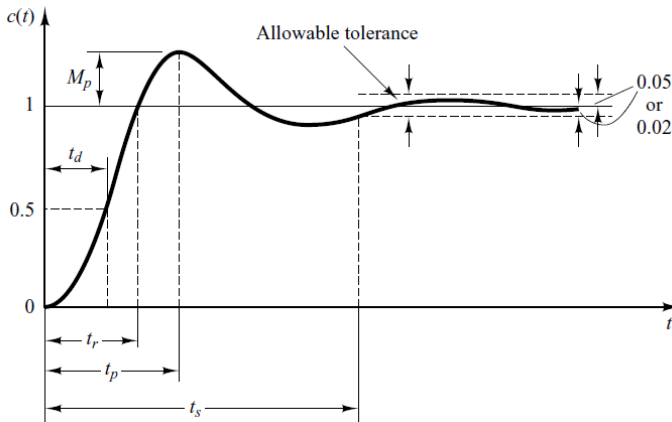
$$M_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

mempunyai *rank* sama dengan n .

2.7 Parameter Performansi Sistem Kendali

Karakteristik performansi dari sebuah sistem kendali dinyatakan dalam beberapa besaran waktu dan nilai *error*. Untuk memudahkan perbandingan respon transien berbagai sistem, hal yang biasa dilakukan adalah menggunakan syarat awal standar bahwa sistem mula-mula dalam keadaan diam, sehingga keluaran dan semua turunan waktunya pada awal respon sama dengan nol. Selanjutnya karakteristik respon secara mudah dapat dibandingkan.

Dalam prakteknya, sistem kendali sering mengalami osilasi sebelum mencapai *steady state*. Dalam menentukan karakteristik respon transien sistem, berikut adalah gambar spesifikasi respon transien pada sebuah grafik yang merepresentasikan sebuah sistem [19]



Gambar 2.3 Spesifikasi respon transien

Beberapa parameter yang terdapat pada gambar 2.3 merepresentasikan:

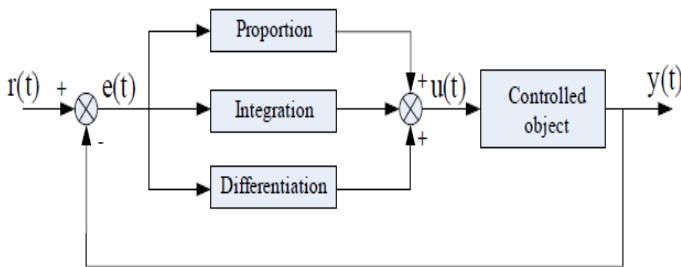
- Delay time* (t_d): waktu yang diperlukan respon untuk mencapai setengah harga akhir yang pertama kali.
- Rise Time* (t_r): waktu yang diperlukan respon untuk naik dari 10 sampai 90%, 5 sampai 95%, atau 0 sampai 100% dari harga akhirnya.
- Peak Time* (t_p): waktu yang diperlukan respon untuk mencapai puncak lewatan yang pertama kali.
- Maximum Overshoot* (M_p): nilai puncak maksimum dari tanggapan diukur dari nilai akhir tanggapan. Biasanya dirumuskan dalam presentase dengan rumus:

$$M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\%$$

- e. *Settling Time* (t_s): waktu yang dibutuhkan tanggapan untuk mencapai nilai akhir dari tanggapan dan tetap berada pada nilai tersebut dalam range presentase tertentu dari nilai akhir (biasanya 2% atau 5%).

2.8 Kontrol PID

Kontrol PID merupakan algoritma pengendali konvensional yang banyak digunakan, PID terdiri dari tiga bagian yaitu proporsional, integral dan derivatif. PID merupakan salah satu *controller* dan *stabilizer* yang sering digunakan di industri yang sederhana dan mudah untuk diimplementasikan[6]. Sistem kendali PID dirancang untuk *Single Input-Single Output* (SISO) sehingga nilai terbaik dari setiap parameter dapat dicari satu per-satu. Selanjutnya, efek umpan balik sistem kendali PID terhadap gerak sistem diperoleh dengan cara memvariasi gain [15]. Gambar diagram blok kontrol PID dapat dilihat pada gambar 2.4



Gambar 2.4 Diagram blok kontrol PID
 algoritma pada pengendali PID dapat dituliskan sebagai berikut:

$$e(t) = r(t) - y(t)$$

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (2.47)$$

dimana $y(t)$ merupakan *output* pada sistem, $r(t)$ adalah *setpoint* pada sistem, $e(t)$ adalah sinyal *error* antara $y(t)$ dan $r(t)$, $u(t)$ merupakan *output* pada pengendali PID, K_p adalah gain *proportional*, T_i adalah waktu konstan *integral* dan T_d adalah waktu konstan *derivativ*. Persamaan (2.47) dapat ditulis menjadi:

$$u(t) = K_p e(t) + K_i \int e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.48)$$

K_i adalah gain *integral*, K_d adalah gain *derivative* dimana $K_i = \frac{K_p}{T_i}$ dan $K_d = K_p T_d$

Setiap parameter pada komponen PID memiliki peran terhadap hasil respon sistem. Masing-masing parameter memiliki fungsi sebagai berikut [16]:

- a. *Proportional* dapat meningkatkan kecepatan respon dan mengontrol akurasi dari sistem. Lebih besar K_p dapat menyebabkan kecepatan respon cepat dan akurasi kontrol yang lebih tinggi. Tetapi jika terlalu besar, *overshoot* akan menjadi besar dan sistem cenderung tak stabil. Sementara itu, jika K_p terlalu kecil akurasi kontrol akan menurun, dan waktu untuk menyesuaikan akan lebih panjang. Kinerja statis dan dinamis akan memburuk.
- b. *Integral* digunakan untuk menghilangkan *error steady-state* pada sistem. Dengan K_i yang besar dapat menghilangkan *error steady-state* lebih cepat. Tapi jika K_i terlalu besar akan menyebabkan *integral saturation* pada permulaan proses kontrol dan *overshoot* besar. Tapi, jika K_i kecil, *error steady-state* akan susah dihilangkan dan akurasi kontrol akan menjadi buruk.
- c. *Derivative* dapat memperbaiki kinerja pada sistem. Parameter ini dapat mencegah dan memprediksi perubahan pada *error* dari segala arah. Tapi jika K_d terlalu besar, respon dari proses akan berhenti lebih

cepat, waktu untuk menyesuaikan akan panjang dan kemampuan menghalangi gangguan pada sistem akan buruk.

Tabel 2.3 Pengaruh *tuning* parameter PID

	t_r	<i>overshoot</i>	t_s	<i>error</i>
Pembesaran K_p	Berkurang	bertambah	Sedikit bertambah	Mengurangi
Pembesaran K_i	Sedikit berkurang	bertambah	bertambah	Mengeliminasi
Pembesaran K_d	Sedikit berkurang	berkurang	Berkurang	Perubahan kecil

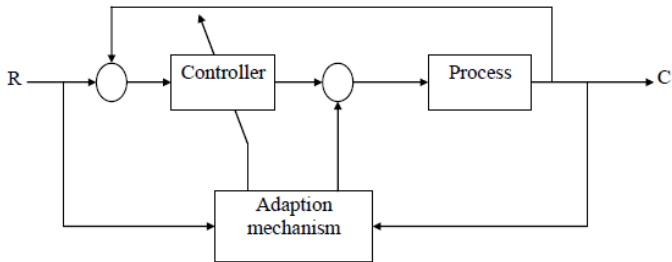
setiap kekurangan dan kelebihan dari pengontrol Proporsional, Integral, Derivatif dapat ditutupi dengan menggabungkan ketiganya secara paralel menjadi pengontrol Proporsional+Integral+Derivatif atau yang bisa disebut dengan pengontrol PID. Secara keseluruhan elemen-elemen PID bertujuan untuk mempercepat reaksi sebuah sistem mencapai *setpoint*-nya, menghilangkan *off set*, menghasilkan perubahan awal yang besar dan mengurangi *overshoot*.

Karakteristik pengontrol PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P, I, dan D. Pengaturan K_p , K_i dan K_d akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari konstanta tersebut dapat diatur lebih menonjol dari pada konstanta yang lain sehingga konstanta tersebut akan memberikan pengaruh pada respon sistem secara keseluruhan[15].

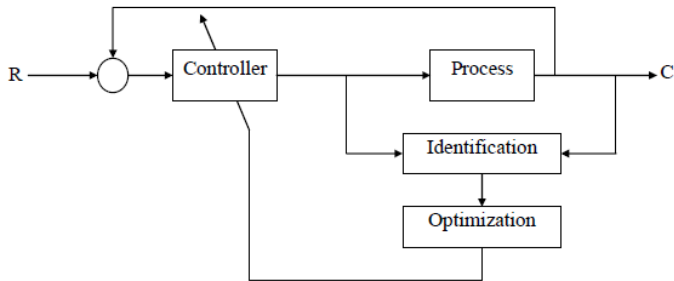
2.9 Kontrol Adaptif

Kontrol adaptif adalah kontrol yang dapat memodifikasi perilaku dalam menanggapi perubahan dinamis dari proses

dan gangguan yang bekerja pada proses. Kontrol adaptif mampu melakukan adjustable parameter dan mekanisme untuk mengatur parameter atau dalam pengertian umumnya berarti mengubah tingkah laku atau karakteristik untuk menyesuaikan diri terhadap keadaan yang baru atau tidak diketahui. Ada dua kategori utama pada kontrol adaptif, *indirect adaptive* (gambar 2.4) dan *direct adaptive* (gambar 2.5). Pada sistem *indirect adaptive* loop dimulai dengan kontrol identifikasi sistem dan menggunakan estimasi parameter untuk estimasi sistem yang sudah diidentifikasi untuk membuat desain kontrol, sedangkan pada sistem *direct adaptive* lebih praktis daripada *indirect adaptive*, *direct adaptive* menggunakan metode estimasi parameter untuk mendapatkan parameter kontroler secara langsung yang selanjutnya akan diproses pada *plant*[17].

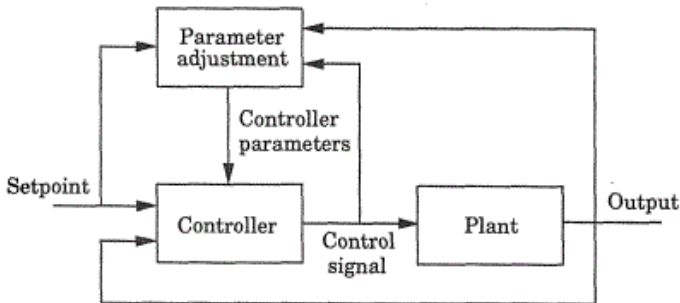


Gambar 2.5 Sistem *direct adaptive*



Gambar 2.6 Sistem *indirect adaptive*

pada umumnya, sistem kontrol adaptif terdiri atas 2 *loop* tertutup, *loop* pertama adalah normal *feedback control* terhadap *plant* dengan kontroler dan *loop* yang kedua adalah *loop* dengan parameter *adjustment loop* [17]. Diagram blok dapat dilihat pada gambar 2.6

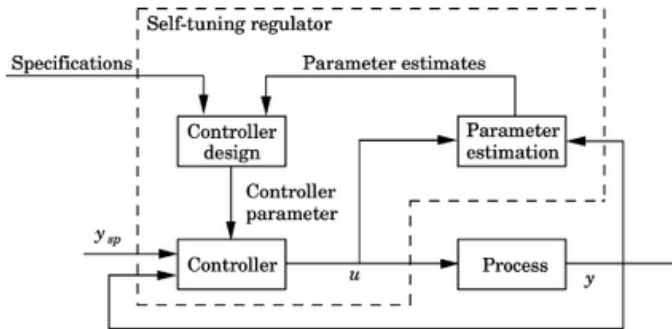


Gambar 2.7 Diagram blok kontrol adaptif

2.10 *Self tuning regulator (STR)*

Kontrol adaptif telah ada sejak tahun 1950 yang diimplementasikan pada autopilot pesawat dan selanjutnya sampai saat ini berkembang teori-teori dan model baru, diantaranya MRAS, *gain sceduling*, *auto tuning*, dan *Self Tuning Regulator (STR)* [6]. Pada tugas akhir ini, kontrol adaptif yang akan digunakan adalah metode PID *Self*

Tuning Regulator (STR). Gambar diagram blok pada *self tuning regulator* ditunjukkan pada gambar 2.7



Gambar 2.8 Diagram blok *self tuning regulator*

Estimator pada blok merepresentasikan proses estimasi parameter secara langsung dengan menggunakan metode *recursive least squares*. Blok Perancangan Pengendali merepresentasikan penyelesaian langsung untuk desain dari parameter yang telah diidentifikasi sebelumnya untuk menghasilkan parameter kontroler terbaru sesuai kondisi objek pada saat itu. Dan terakhir pada blok *controller* adalah untuk menghitung aksi kontrol yang akan diberikan kepada objek dengan parameter kontroler yang telah dihitung pada blok sebelumnya. Sehingga sistem dapat dikatakan sebagai otomatisasi. Yang mana model dari proses dan desain kontrol diperbaharui setiap saat [17].

2.11 Recursive Least Squares

Pada kendali adaptif algoritma *recursive least squares* merupakan salah satu estimator yang sering digunakan untuk estimasi parameter kendali. Pada Tugas Akhir ini RLS digunakan untuk mengestimasi parameter K_p , K_i dan K_d agar parameter PID mampu memperbarui saat sistem mengalami perubahan sewaktu-waktu, hasil dari algoritma

recursive least squares selanjutnya digunakan pada blok kontroler untuk menghasilkan *gain* kendali yang mampu memperbarui secara *online*.

Metode *least squares* pertama kali dirumuskan untuk menentukan orbit planet pada akhir abad ke delapan belas oleh Gauss. Prinsip *least squares* adalah meminimalkan perbedaan antara pengamatan sebenarnya dan nilai penrhitungan. *Least squares* dapat diterapkan pada model matematika pada persamaan:

$$\begin{aligned} y(t) &= \varphi_1(t)\theta_1 + \varphi_2(t)\theta_2 + \dots + \varphi_n(t)\theta_n \\ &= \varphi(t)^T \theta \end{aligned} \quad (2.49)$$

dengan y adalah variabel yang diamati, $\theta_1, \theta_2, \dots, \theta_n$ parameter yang tak-diketahui, dan $\varphi_1, \varphi_2, \dots, \varphi_n$ adalah fungsi yang diketahui yang bergantung variabel diketahui yang lain. Variabel φ_i adalah variabel regresi atau *regressor*. Model persamaan (2.50) juga disebut model regresi dengan vektor:

$$\varphi^T(t) = [\varphi_1(t) \quad \varphi_2(t) \quad \dots \quad \varphi_n(t)]$$

$$\theta^T = [\theta_1 \quad \theta_2 \quad \dots \quad \theta_n]$$

Diasumsikan bahwa $\hat{\theta}$ nilai estimasi parameter tak-diketahui, dan perbedaan antara keluaran sebenarnya dan model estimasi adalah

$$y(t) - \hat{y}(t) = y(t) - \varphi^T(t)\hat{\theta}$$

disebut sebagai *error* keluaran model biasa disimbolkan $\varepsilon(t)$, sehigga:

$$y(t) = \varphi^T(k)\hat{\theta} + \varepsilon(t)$$

error dari *least-squares* dapat ditulis sebagai berikut:

$$V(\theta, t) = \frac{1}{2} \sum_{i=1}^t (y(i) - \varphi^T(i)\hat{\theta})^2$$

metode *least squares* digunakan untuk memperoleh solusi minimal pada fungsi $V(\theta, t)$

Dalam kendali adaptif pengamatan diperoleh berurutan secara *real time*. Sehingga memerlukan untuk perhitungan secara rekursif untuk menghemat waktu komputasi. Konsep rekursif dapat digunakan sehingga nilai yang diperoleh pada waktu $t - 1$ dapat digunakan untuk memperoleh estimasi pada waktu t . Nilai prediksi *error* dapat dirumuskan pada persamaan (2.50)

$$\varepsilon(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1) \quad (2.50)$$

dan untuk menentukan estimasi parameter pada RLS *with forgetting factor* untuk menentukan $\hat{\theta}(t)$ digunakan persamaan:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t-1)\varepsilon(t) \quad (2.51)$$

$$\begin{aligned} K(t) &= P(t)\varphi(t) \\ &= P(t-1)\varphi(t)(\lambda I + \varphi^T(t)P(t-1)\varphi(t))^{-1} \end{aligned} \quad (2.52)$$

$$P(t) = (I - K(t)\varphi^T(t))P(t-1)/\lambda \quad (2.53)$$

persamaan (2.51) merupakan nilai estimasi, persamaan (2.52) merupakan *weighting factor* atau gain estimasi dan persamaan (2.53) merupakan matriks kovarian, λ merupakan variabel *forgetting factor* dengan nilai $0 < \lambda \leq 1$. Nilai estimasi $\hat{\theta}(t)$ diperoleh dengan menambahkan koreksi pada estimasi sebelumnya $\hat{\theta}(t-1)$, yang selanjutnya nilai estimasi parameter pada waktu t digunakan pada *plant* sistem.

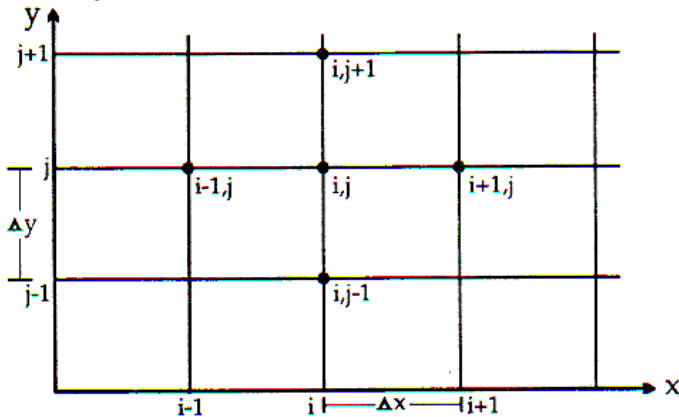
Untuk melakukan inisialisasi pada algoritma RLS, inisialisasi yang diberikan adalah parameter kendali $\hat{\theta}(0)$ dan inisialisasi matriks kovarian $P(0)$, dengan rumus:

$$P(0) = \gamma I \quad (2.54)$$

dimana $\gamma > 0$ dan I merupakan matriks identitas $n \times n$. Sedangkan untuk inisialisasi nilai parameter θ_0 adalah dengan menentukan nilai konstanta sebarang dengan $\theta \in \mathbb{R}^n$, dan untuk φ adalah matriks berukuran $n \times 1$.

2.12 Metode Beda Hingga

Untuk mengimplementasikan model kontinu diubah ke bentuk diskrit diperlkan metode pendekatan beda hingga. Perhatikan gambar 2.9 berikut:



Gambar 2.9 Jaringan titik metode beda hingga

Gambar 2.9 merupakan jaringan titik hitungan pada bidang x - y yang dapat dibagi menjadi sejumlah pias segi empat dengan sisi Δx dan Δy . Panjang pias dalam arah x adalah Δx dan dalam arah y adalah Δy . Dengan menggunakan jaringan titik hitungan pada gambar semua diferensial ditulis pada titik hitungan (i,j) . Bentuk

pendekatan untuk turunan pertama dibagi menjadi tiga yaitu beda maju, beda mundur dan beda pusat, dengan pendekatan ditunjukkan pada persamaan (2.55) sampai (2.57)

- a. Pendekatan beda maju

$$\frac{du}{dx} \approx \frac{u_{i+1,j} - u_{i,j}}{\Delta x} \quad (2.55)$$

- b. Pendekatan beda mundur

$$\frac{du}{dx} \approx \frac{u_{i,j} - u_{i-1,j}}{\Delta x} \quad (2.56)$$

- c. Pendekatan beda pusat

$$\frac{du(x_j)}{dx} \approx \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} \quad (2.57)$$

Halaman ini sengaja dikosongkan

BAB III

METODE PENELITIAN

Dalam bab ini diuraikan langkah-langkah sistematis yang dilakukan dalam proses pengerjaan Tugas Akhir. Kegiatan penelitian dalam Tugas Akhir ini terdiri atas: studi literatur, kajian model kendali haluan kapal SSV BRP TARLAC (LD-601), perancangan desain kendali haluan kapal menggunakan PID adaptif, simulasi dan analisis hasil, dan penarikan kesimpulan.

3.1 Studi Literatur

Pada tahap ini dipelajari literatur-literatur ilmiah yang memiliki hubungan dan mendukung topik penelitian PID adaptif meliputi:

- a. Kendali PID
- b. Pemodelan matematika kendali haluan kapal SSV BRP TARLAC (LD-601)
- c. *Online tuning parameter* menggunakan RLS
- d. Metode Beda Hingga
- e. Keterkontrolan dan keteramatan
- f. Parameter Performansi
- g. Kendali adaptif.

3.2 Persamaan Keadaan Model Matematika Kendali Haluan Kapal SSV BRP TARLAC (LD-601)

Pada tahap ini dilakukan identifikasi model kendali haluan kapal SSV BRP TARLAC (LD-601) dengan model matematika menggunakan pendekatan model Nomoto orde 2, dimana hasilnya berupa fungsi transfer dengan parameter-parameter yang berasal dari penurunan rumus dinamika kapal, dan selanjutnya dari fungsi transfer:

$$\frac{r(s)}{\delta_R(s)} = \frac{244,2 \cdot s + 484,6}{0,1203 \cdot s^2 + 0.7312 \cdot s + 1}$$

Dengan menggunakan invers transformasi Laplace persamaan kendali haluan kapal dapat diubah ke dalam bentuk sistem persamaan diferensial yang selanjutnya akan diperoleh bentuk persamaan ruang keadaan/*state space* sebagai berikut:

$$\dot{x} = \mathbf{A}x(t) + \mathbf{B}u(t)$$

dan dengan mengambil keluaran sistem adalah sudut yaw (ψ) maka keluaran sistem adalah:

$$y = \mathbf{C}x + \mathbf{D}$$

Dengan:

A : Matriks keadaan

B : Matriks masukan

C : Matriks keluaran

D : Matriks transmisi langsung

Selanjutnya, dengan menggunakan teorema 2.5 akan dilakukan analisa kestabilan sistem. Selanjutnya sistem persamaan akan menjadi *plant* pada desain kontrol menggunakan PID adaptif untuk mengendalikan sudut yaw. Namun sebelum dirancang pada desain kendali persamaan keadaan harus melalui uji keterkontrolan dan keteramatan dengan menghitung:

$$\text{rank } M_c = \text{rank } M_o = n$$

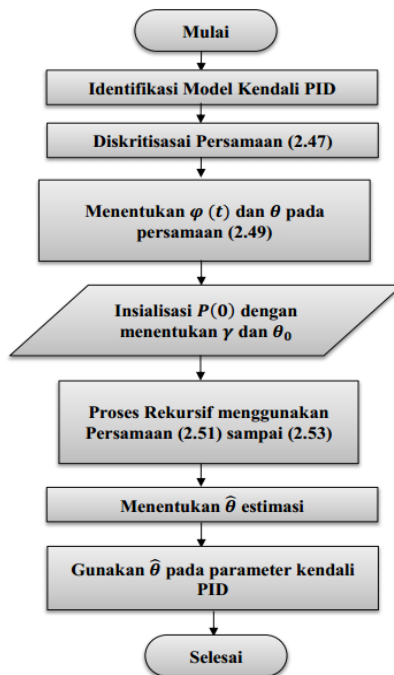
3.3 Desain Kendali Haluan Kapal menggunakan PID adaptif

Pada tahap ini akan dilakukan perancangan desain kendali PID adaptif. Tahap pertama adalah identifikasi parameter, selanjutnya desain *plant* haluan kapal SSV, perancangan kendali PID konvensional, dengan menentukan gain K_p, K_i, K_d , setelah itu identifikasi model matematika kontrol PID ke bentuk model regresi linier,

setelah itu dari model PID diskrit diperoleh parameter θ dan φ , selanjutnya, menggunakan algoritma *recursive least squares* untuk estimasi parameter PID yang meliputi:

1. Insialisasi parameter kendali $\hat{\theta}(0) = \theta(0) = \theta_0$ dan parameter γ untuk memperoleh $P(0)$ menggunakan persamaan (2.54)
2. Hitung $\hat{\theta}(t)$ dengan menggunakan persamaan (2.51) sampai (2.53)
3. Implementasikan hasil estimasi parameter pada blok kontrol.

Berikut adalah diagram alir untuk desain kendali PID adaptif



Gambar 3.1 Diagram alir kendali PID adaptif

3.4 Simulasi dan Penarikan Kesimpulan

Pada tahap ini, simulasi dilakukan dengan menggunakan *simulink* pada *software* MATLAB. Dari grafik simulasi merepresentasikan respon kendali PID konvensional dan kendali PID adaptif terhadap *setpoint* berupa sudut *yaw* yang diberikan. Selanjutnya pengamatan respon kendali PID konvensional dan PID adaptif dari keluaran sistem. Spesifikasi respon transien pengendali yang diamati antara lain [19]:

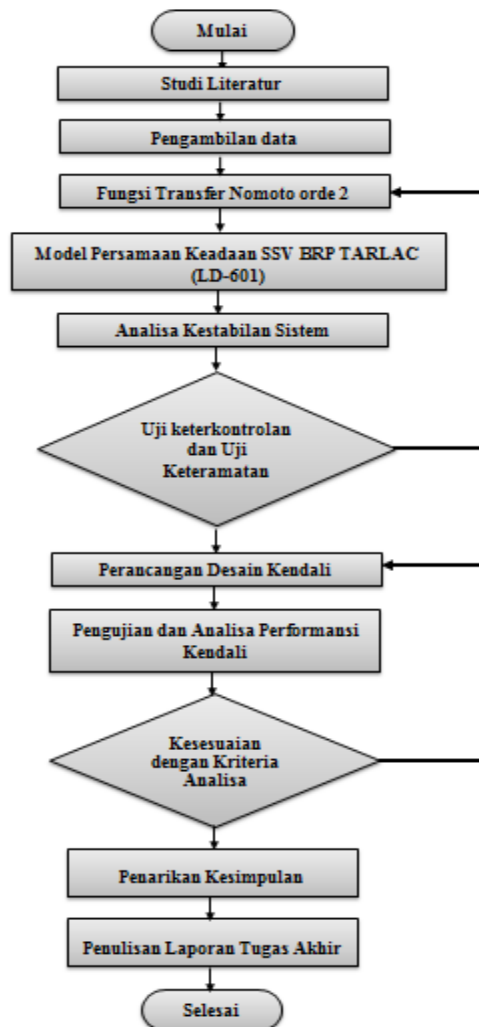
- a. Waktu Tunda / *delay time* (t_d)
- b. Waktu Naik / *rise time* (t_r)
- c. Waktu Puncak / *peak time* (t_p)
- d. Persen lewatan maksimum / *maximum overshoot* (M_p)
- e. Waktu Penetapan / *Settling time* (t_s)

Setelah melakukan pengamatan respon transien pengendali, tahap selanjutnya melakukan penarikan kesimpulan dari hasil simulasi dengan membandingkan performansi pengendali PID konvensional dan PID adaptif.

3.5 Penulisan Laporan Tugas Akhir

Pada tahap ini, dilakukan penulisan Tugas Akhir setelah mendapatkan simulasi dan penarikan kesimpulan dan mendapatkan jawaban dari masalah dari topik.

Berikut adalah gambar diagram alir metode penelitian pada Tugas Akhir ini:



Gambar 3.2 Diagram alir metode penelitian

Halaman ini sengaja dikosongkan

BAB IV ANALISIS DAN PEMBAHASAN

Pada bab ini dijelaskan tentang model persamaan keadaan kendali haluan kapal (*ship heading control*) untuk model kapal SSV BRP TARLAC (LD-601) dan akan dilakukan analisis kestabilan sistem yang selanjutnya akan digunakan dalam perancangan sistem kendali, selanjutnya dilakukan uji keterkontrolan dan keteramatan sistem. Selanjutnya hasil perancangan desain kendali yang telah dibuat disimulasikan menggunakan *simulink* MATLAB. Selanjutnya dari hasil simulasi dilakukan analisis untuk mengetahui karakteristik respon kendali pada gerak *yaw* untuk kapal SSV BRP TARLAC (LD-601).

4.1 Persamaan Keadaan Kendali Haluan Kapal

Berdasarkan fungsi transfer pada persamaan (2.36) dapat direpresentasikan dari fungsi transfer ke persamaan keadaan $\dot{\mathbf{x}}(t) = \mathbf{A} \mathbf{x}(t) + \mathbf{B} u(t)$. dengan menambahkan $x(s) = \psi(s)$ persamaan (2.36) menjadi,

$$\frac{r(s)}{\delta_R(s)} = \frac{(244,2.s + 484,6)}{(0,1203.s^2 + 0,7312.s + 1)} \frac{x(s)}{x(s)} \quad (4.1)$$

jika,

$$\delta_R(s) = (0,1203.s^2 + 0,7312.s + 1)x(s) \quad (4.2)$$

dan

$$r(s) = (244,2.s + 484,6)x(s) \quad (4.3)$$

misalkan, $r(s) = Y(s)$ sebagai keluaran sistem dan $\delta_R(s) = U(s)$ sebagai masukan sistem, maka fungsi transfer pada persamaan (4.2) dapat diturunkan menjadi,

$$x(s) = \frac{1}{0,1203.s^2 + 0,7312.s + 1} U(s) \quad (4.4)$$

dan

$$Y(s) = (244,2.s + 484,6) x(s) \quad (4.5)$$

sehingga, dari persamaan (4.2) diperoleh:

$$U(s) = (0,1203.s^2 + 0,7312.s + 1) x(s) \quad (4.6)$$

selanjutnya, dengan menggunakan invers transformasi Laplace dengan asumsi $x(0) = 0$, persamaan (4.6) akan diubah ke domain waktu t

$$\begin{aligned} \mathcal{L}^{-1}\{0,1203.x(s)s^2 + 0,7312.x(s)s + x(s)\} \\ = \mathcal{L}^{-1}\{U(s)\} \end{aligned} \quad (4.7)$$

dari persamaan diatas domain diubah ke domain waktu

$$0,1203\ddot{x}(t) + 0,7312\dot{x}(t) + x(t) = U(t) \quad (4.8)$$

dari persamaan diferensial diatas akan diubah ke dalam bentuk *state-space* dengan memisalkan

$$\begin{aligned} x_1 &= x(t) \\ x_2 &= \dot{x}_1 = \dot{x}(t) \\ \dot{x}_2 = \ddot{x}_1 = \ddot{x}(t) &= -6,08\dot{x}(t) - 8,31x(t) + 8,31U(t) \end{aligned}$$

sehingga diperoleh persamaan ruang keadaan

$$\dot{x} = \mathbf{A}x + \mathbf{B}u$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -8,31 & -6,08 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 8,31 \end{bmatrix} u(t) \quad (4.9)$$

Dimana $x_1 = x(t) = \psi(t)$ merupakan sudut *yaw* dan $x_2 = \dot{x}_1 = \dot{x}(t) = \dot{\psi}(t)$ merupakan kecepatan sudut *yaw*. Persamaan (4.9) dapat ditulis kembali menjadi

$$\begin{bmatrix} \dot{\psi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -8,31 & -6,08 \end{bmatrix} \begin{bmatrix} \psi \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ 8,31 \end{bmatrix} \delta_R \quad (4.10)$$

karena pada Tugas Akhir ini, keluaran yang akan dikontrol adalah sudut *yaw* (ψ) dan mengabaikan kecepatan *yaw* (r), maka persamaan keluaran sistem adalah $y = Cx$ dengan matriks:

$$C = [1 \quad 0]$$

sehingga diperoleh persamaan keluaran untuk model haluan kapal SSV BRP TARLAC (LD-601):

$$y = [1 \quad 0] \begin{bmatrix} \psi \\ r \end{bmatrix} \quad (4.11)$$

Jadi, diperoleh matriks

$$A = \begin{bmatrix} 0 & 1 \\ -8,31 & -6,08 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 8,31 \end{bmatrix}$$

$$C = [1 \quad 0]$$

4.1.1 Analisa Kestabilan Sudut *yaw*

Stabil merupakan suatu kondisi sistem yang jika mengalami gangguan dari dalam maupun luar mampu kembali ke kondisi titik kesetimbangan. Dalam menganalisa kestabilan dapat dilihat dari bentuk matriks *state-space* gerak *yaw* pada persamaan (4.10), dengan matriks **A** adalah:

$$A = \begin{bmatrix} 0 & 1 \\ -8,31 & -6,08 \end{bmatrix}$$

Kemudian, dicari akar-akar karakteristik (nilai eigen λ) dengan rumus $\det(\lambda I - A) = 0$, diperoleh:

$$\begin{vmatrix} \lambda & 0 \\ 0 & \lambda \end{vmatrix} - \begin{vmatrix} 0 & 1 \\ -8,31 & -6,08 \end{vmatrix} = 0$$

$$\begin{vmatrix} \lambda & -1 \\ 8,31 & \lambda + 6,08 \end{vmatrix} = 0$$

Sehingga diperoleh persamaan karakteristik λ sebagai berikut:

$$p(\lambda) = \lambda^2 + 6.08 \lambda + 8.3 = 0$$

$$\lambda_{1,2} = -\frac{6.08}{2} \pm \frac{\sqrt{(6.08)^2 - 4(8.31)}}{2}$$

sehingga diperoleh

$$\lambda_1 = -3.040 + 0.965 = -2.075 \text{ dan}$$

$$\lambda_2 = -3.040 - 0.965 = -4.005$$

Menurut teorema 2.5.1 sistem dikatakan stabil asimtotik karena $\text{Re}(\lambda_i) < 0$ untuk $i = 1, 2$

4.1.2 Uji Keterkontrolan dan Keteramatan

Sebelum merancang sebuah pengendali, sistem harus diketahui terlebih dahulu keterkontrolan dan keteramatannya. Keterkontrolan dan keteramatan mempunyai peran penting dalam perancangan pengendali, uji keterkontrolan dan keteramatan ini menentukan bisa tidaknya dilanjutkan perancangan sistem kendali.

1. Uji Keterkontrolan

Matriks keterkontrolan diberikan oleh persamaan berikut:

$$M_c = [B \quad AB]$$

matriks keterkontrolan untuk *input* δ_R adalah:

$$M_c = \begin{bmatrix} 0 & 8.314 \\ 8.314 & -50.547 \end{bmatrix}$$

2. Uji Keteramatan

Matriks keteramatan diberikan oleh persamaan berikut:

$$M_o = \begin{bmatrix} C \\ CA \end{bmatrix}$$

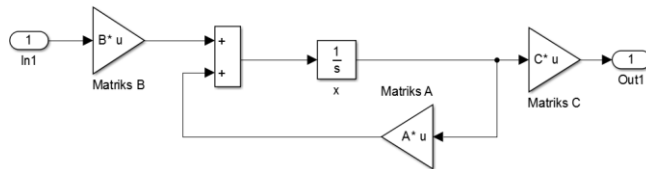
Matriks keteramatan diberikan oleh persamaan berikut:

$$M_o = \begin{bmatrix} 1 & 0 \\ -8.314 & -6.07 \end{bmatrix}$$

Dengan menggunakan bantuan *software* MATLAB maka diperoleh $\text{rank}(M_c) = 2$ dan $\text{rank}(M_o) = 2$. Nilai rank dari matriks M_c dan M_o sama dengan orde sistem, sehingga sistem dapat dikatakan terkontrol dan teramati. Dengan demikian dapat dilakukan perancangan sistem kendali.

4.2 Perancangan Desain Kendali PID Konvensional

Pada sub bab ini, akan dibahas perancangan desain kendali PID konvensional untuk kendali sudut *heading* kapal SSV BRP TARLAC (LD-601) agar tetap stabil sesuai dengan nilai *setpoint* sudut *heading* yang diberikan. Sebelum merancang desain kendali terlebih dahulu merancang *plant* yang akan dikendalikan yaitu sudut *heading* dengan menggunakan persamaan keadaan kapal SSV BRP TARLAC (LD-601) pada persamaan (4.10). Diagram blok persamaan keadaan sudut *heading* disajikan pada gambar 4.1.



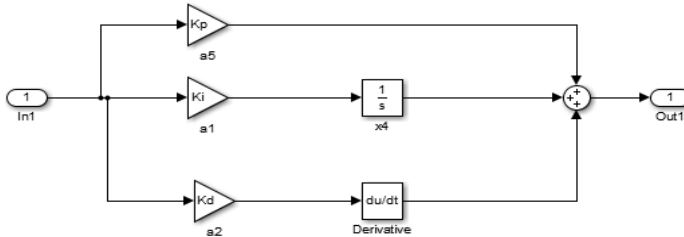
Gambar 4.1 *plant* sudut *heading* kapal

Matriks **A** dan **B** pada *plant* merupakan matriks persamaan keadaan gerak *yaw* kapal SSV BRP TARLAC (LD-601) yang diperoleh pada persamaan (4.10) dan matriks **C** merupakan keluaran sistem pada persamaan (4.11) dengan menggunakan *plant* pada gambar 4.1 sistem kendali yang dirancang diharapkan mampu untuk mengendalikan sudut *heading* agar mampu mencapai *setpoint* yang diberikan.

Selanjutnya, dengan menggunakan *plant* pada gambar 4.1 akan dirancang desain kendali PID konvensional. PID merupakan sebuah sistem kendali yang mempunyai tiga komponen utama yaitu Proporsional, Integrator, Derivatif dengan tiga *gain* K_p, K_i, K_d . Untuk merancang sistem kendali PID terlebih dahulu menentukan *gain* K_p, K_i, K_d yang sesuai pada persamaan:

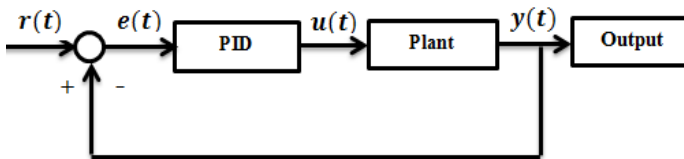
$$u(t) = K_p e(t) + K_i \int e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (4.12)$$

dengan $e(t)$ adalah selisih antara nilai pengukuran dan *setpoint*. Desain kendali PID konvensional ditunjukkan pada gambar 4.2



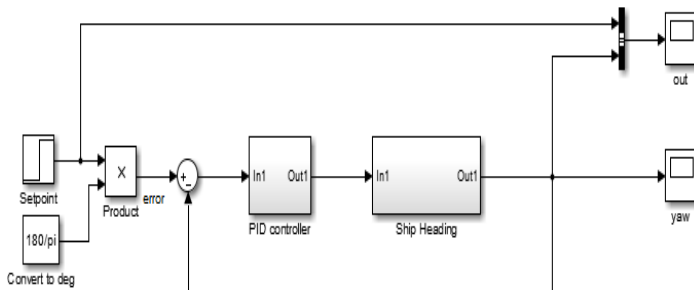
Gambar 4.2 Desain kendali PID

Parameter pada kendali PID konvensional diperoleh dengan cara *trial and error* sampai mendapatkan kinerja *gain* yang sesuai dengan yang diinginkan. Setelah mendapatkan *gain* pengendali PID selanjutnya akan dilakukan simulasi dengan menggunakan *simulink* MATLAB. Diagram blok yang dirancang untuk simulasi sudut *heading* ditampilkan pada gambar 4.3.



Gambar 4.3 Diagram blok PID Konvensional

Pada gambar 4.3 $r(t)$ merupakan *setpoint* yang diberikan pada sistem, $e(t)$ merupakan *error* kendali PID konvensional yaitu selisih antara keluaran sistem atau *heading* $y(t)$ dan $r(t)$ atau dapat dirumuskan $e(t) = y(t) - r(t)$. Sinyal dari keluaran kendali PID disimbolkan $u(t)$ yang akan digunakan pada *plant* haluan kapal. Perancangan desain kendali PID dengan menggunakan Simulink MATLAB ditunjukkan pada gambar 4.4.



Gambar 4.4 Desain Kendali PID konvensional

4.3 Perancangan Desain Kendali PID adaptif

Pada subbab ini akan dibahas perancangan desain kendali PID adaptif untuk haluan kapal SSV BRP TARLAC (LD-601) agar parameter pengendali PID mampu beradaptasi setiap waktu saat sistem diberikan gangguan dan tetap mampu mencapai *setpoint* yang diharapkan. Sebelum merancang desain kendali PID adaptif tahap pertama adalah menentukan model diskrit kendali haluan kapal, selanjutnya identifikasi parameter berdasarkan persamaan kendali PID, selanjutnya proses *online tuning* parameter menggunakan RLS dan selanjutnya perancangan desain kendali PID adaptif.

4.3.1 Sistem Diskrit Haluan Kapal SSV BRP TARLAC (LD-601)

Sebelum melakukan perancangan kendali PID adaptif, *plant* yang akan dikontrol merupakan model diskrit, sehingga perlu dilakukan proses diskritisasi persamaan (4.9). Diberikan *plant* pada sistem kontinu kendali haluan kapal SSV BRP TARLAC (LD-601) sebagai berikut:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -8,31 & -6,08 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 8,31 \end{bmatrix} u(t)$$

dengan mendekati $\dot{x}_1(t), \dot{x}_2(t)$ dengan skema beda maju pada persamaan (2.55) diperoleh:

$$\begin{bmatrix} \frac{x_1(k+1) - x_1(k)}{\Delta t} \\ \frac{x_2(k+1) - x_2(k)}{\Delta t} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -8,31 & -6,08 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 8,31 \end{bmatrix} u(k)$$

selanjutnya kedua ruas dikalikan Δt sehingga,

$$\begin{bmatrix} x_1(k+1) - x_1(k) \\ x_2(k+1) - x_2(k) \end{bmatrix} = \begin{bmatrix} 0 & \Delta t \\ -8,31\Delta t & -6,08\Delta t \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 8,31\Delta t \end{bmatrix} u(k)$$

selanjutnya kedua ruas dijumlah dengan $\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$ sehingga diperoleh,

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & \Delta t \\ -8,31\Delta t & -6,08\Delta t \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 8,31\Delta t \end{bmatrix} u(k)$$

atau dapat ditulis menjadi:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ -8,31\Delta t & 1 - 6,08\Delta t \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 8,31\Delta t \end{bmatrix} u(k)$$

dengan mengambil waktu *sampling* $\Delta t = 0,1$ diperoleh:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0,1 \\ -0,831 & 0,392 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0,831 \end{bmatrix} u(k)$$

atau dapat ditulis sebagai persamaan

$$x(k+1) = \mathbf{A} x(k) + \mathbf{B} u(k) \quad (4.13)$$

dengan

$$\mathbf{A} = \begin{bmatrix} 1 & 0,1 \\ -0,831 & 0,392 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 0 \\ 0,831 \end{bmatrix}$$

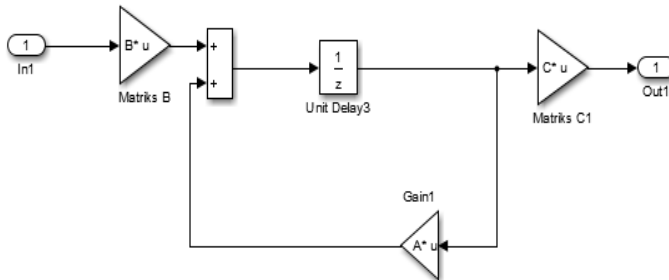
dengan mengambil keluaran sistem adalah sudut *heading* diperoleh persamaan keluaran

$$y = \mathbf{C} x(k) \quad (4.14)$$

dengan matriks

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

selanjutnya, persamaan (4.13) dan (4.14) digunakan pada *plant* sistem kendali PID adaptif. Perancangan *plant* untuk kendali PID adaptif ditunjukkan pada gambar 4.5.



Gambar 4.5 *Plant* sistem diskrit

pada gambar 4.5 matriks keadaan dan matriks keluaran diperoleh pada persamaan (4.13) dan (4.14) dengan keluaran sistem adalah sudut *heading*.

4.3.2 Identifikasi Parameter Kendali PID

Pada perancangan kendali PID adaptif *gain* kontrol PID harus mampu beradaptasi ketika sistem mengalami perubahan, sehingga perlu dilakukan identifikasi parameter PID, yang selanjutnya akan dilakukan proses *tuning* secara *online* menggunakan *recursive least squares*. Identifikasi parameter PID berdasarkan persamaan PID konvensional

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(\tau) d\tau + T_d \frac{de(t)}{dt} \right]$$

selanjutnya model kendali PID kontinu akan diubah ke bentuk diskrit, dengan cara mencari model diskrit masing-masing penyusun PID yang ditunjukkan pada persamaan (4.14)

$$u(k) = u_p(k) + u_i(k) + u_d(k) \quad (4.14)$$

dengan $u_p(k)$ adalah model diskrit penyusun gain proporsional, berdasarkan model kontinu dari proporsional

$$u_p(t) = K_p e(t)$$

karena bentuk persamaanya statis, model diskrit dapat diperoleh langsung tanpa menggunakan pendekatan apapun. Model diskrit dari proporsional dapat dilihat pada persamaan (4.15)

$$u_p(k) = K_p e(k) \quad (4.15)$$

berdasarkan persamaan (4.15) terlihat bahwa keluaran kendali proporsional untuk waktu cuplik ke- k pada dasarnya hanya bergantung pada *error* saat itu juga.

Untuk suku kontrol integral diberikan persamaan kontinu:

$$u_i(t) = \frac{K_p}{T_i} \int e(\tau) d\tau$$

karena suku kontrol integral memiliki persamaan yang dinamis, maka model diskrit yang dapat digunakan adalah dengan menggunakan pendekatan numerik. Pada Tugas akhir ini metode numerik yang digunakan adalah metode *backward rectangular*. Pendekatan untuk kontrol integral diberikan pada persamaan :

$$u_i(k) = \frac{K_p}{T_i} \sum_{i=0}^k e(i) T_s$$

atau dapat ditulis:

$$u_i(k) = \frac{K_p T_s}{T_i} \sum_{i=0}^k e(i) \quad (4.16)$$

terlihat bahwa hasil pendekatan kontrol integral selain bergantung dengan *error* pada waktu $-k$ juga bergantung *error* sebelumnya, dengan T_s adalah waktu *sampling*.

Sedangkan untuk kontrol diferensial diberikan model kontinu:

$$u_d(t) = K_p T_d \frac{de(t)}{dt}$$

dengan menggunakan pendekatan metode *backward difference* pada persamaan (2.56) keluaran pada waktu ke- k dapat dihitung berdasarkan selisih *error* saat itu dengan *error* waktu sebelumnya, dapat ditunjukkan pada persamaan berikut:

$$u_d(k) = K_p T_d \frac{e(k) - e(k-1)}{T_s} \quad (4.17)$$

Sehingga PID diskrit dapat dirumuskan dengan menjumlahkan persamaan (4.15), (4.16), (4.17), sehingga diperoleh model PID diskrit sebagai berikut:

$$u(k) = K_p \left[e(k) + \frac{T_s}{T_i} \sum_{i=0}^k e(i) + T_d \frac{e(k) - e(k-1)}{T_s} \right] \quad (4.18)$$

selanjutnya, hitung persamaan $u(k) - u(k-1)$, sehingga

$$\begin{aligned} u(k) - u(k-1) = & K_p \left[e(k) + \frac{T_s}{T_i} \sum_{i=0}^k e(i) \right. \\ & \left. + T_d \frac{e(k) - e(k-1)}{T_s} \right] \\ & - K_p \left[e(k-1) + \frac{T_s}{T_i} \sum_{i=0}^{k-1} e(i) \right. \\ & \left. + T_d \frac{e(k-1) - e(k-2)}{T_s} \right] \end{aligned}$$

atau dapat ditulis menjadi persamaan:

$$\begin{aligned} u(k) - u(k-1) = & K_p e(k) + \frac{K_p T_s}{T_i} e(k) + \frac{K_p T_d}{T_s} e(k) \\ & - \frac{K_p T_d}{T_s} e(k-1) - K_p e(k-1) \\ & - \frac{K_p T_d}{T_s} e(k-1) + \frac{K_p T_d}{T_s} e(k-2) \end{aligned}$$

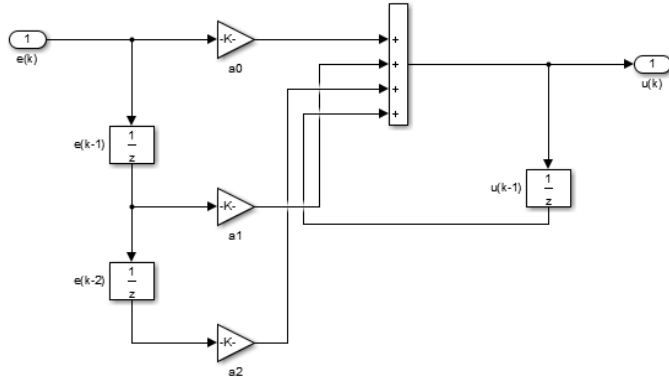
dengan memisahkan elemen $e(k), e(k-1), e(k-2)$ diperoleh persamaan:

$$\begin{aligned} u(k) - u(k-1) = & K_p \left(1 + \frac{T_s}{T_i} + \frac{T_d}{T_s} \right) e(k) \\ & + K_p \left(-1 - 2 \frac{T_d}{T_s} \right) e(k-1) + \frac{K_p T_d}{T_s} e(k-2) \end{aligned}$$

dengan $e(k)$ adalah perbandingan antara *setpoint* dan nilai pengukuran atau dapat dirumsukan $e(k) = r(k) - y(k)$. Sehingga model kendali PID dapat ditulis menjadi:

$$u(k) = u(k-1) + a_0 e(k) + a_1 e(k-1) + a_2 e(k-2) \quad (4.19)$$

desain kendali PID pada persamaan (4.19) ditunjukkan pada gambar 4.6



Gambar 4.6 Perancangan PID diskrit

pada gambar 4.6 nilai parameter-parameter diberikan pada persamaan:

$$a_0 = K_p \left(1 + \frac{T_s}{T_i} + \frac{T_d}{T_s} \right) \quad (4.20)$$

$$a_1 = K_p \left(-1 - 2 \frac{T_d}{T_s} \right) \quad (4.21)$$

$$a_2 = \frac{K_p T_d}{T_s} \quad (4.22)$$

terlihat bahwa model PID pada persamaan (4.19) bergantung pada nilai $u(k-1)$, $e(k)$, $e(k-1)$, $e(k-2)$.

Dengan menggunakan persamaan (2.19) diperoleh model regresi linear

$$u(k) = \varphi^T(k)\theta$$

dengan parameter θ diperoleh dari persamaan (4.20)-(4.22), dan *regressor* atau $\varphi(k)$ adalah matriks berukuran $n \times 1$ dengan elemen didalamnya adalah $e(k), e(k-1), e(k-2)$, sehingga diperoleh parameter θ^T dan $\varphi^T(k)$ pada persamaan (4.23) dan (4.24):

$$\theta^T = [a_0 \ a_1 \ a_2 \ a_3] \quad (4.23)$$

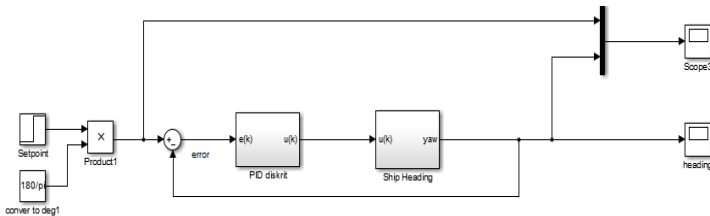
$$\varphi^T(k) = [e(k) \ e(k-1) \ e(k-2) \ u(k-1)] \quad (4.24)$$

Model PID juga dapat direpresentasikan sebagai fungsi transfer diskrit PID diperoleh dengan menggunakan transformasi z berdasarkan persamaan (4.19) diperoleh fungsi transfer pada persamaan (4.25):

$$u(z)(1 - z^{-1}) = e(z)(a_0 + a_1 z^{-1} + a_2 z^{-2})$$

$$\frac{u(z^{-1})}{e(z^{-1})} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - z^{-1}} \quad (4.25)$$

perancangan desain kendali PID diskrit menggunakan simulink ditunjukkan pada gambar 4.7.



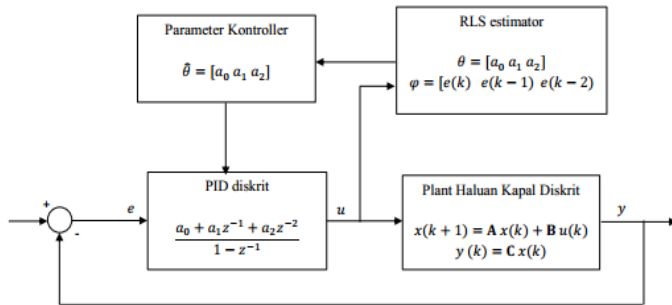
Gambar 4.7 Kendali PID diskrit

4.3.3 Online tuning menggunakan RLS

Algoritma proses *online tuning* pada parameter PID diberikan:

- I. Inisialisasi parameter kendali θ_0 menggunakan parameter kendali PID konvensional dan $P(0)$ menggunakan persamaan (2.54)
- II. Menetapkan matriks $\varphi(k)^T$ dan θ menggunakan persamaan (2.49)
- III. Tentukan $\hat{\theta}(t)$ dengan menggunakan persamaan (2.50) sampai (2.53)
- IV. Hasil estimasi pada tahap 3 akan diterapkan pada parameter kendali PID.

Diagram blok untuk proses RLS ditunjukkan pada gambar 4.8.



Gambar 4.8 Diagram Blok RLS

pada gambar 4.8 menunjukkan skema proses *online tuning* menggunakan RLS, variabel *regressor* dan parameter diperoleh dengan identifikasi model PID diskrit, dan hasil estimasi parameter akan digunakan untuk merancang kendali PID. Berikut adalah beberapa tahapan dalam

menentukan parameter estimasi menggunakan algoritma *recursive least squares*.

a. Tahap inisialisasi

Pada tahap ini akan dilakukan inisialisasi awal untuk parameter yang akan diestimasi θ_0 dan matriks kovarian $P(0)$. Untuk inisialisasi parameter diperoleh dengan menggunakan parameter PID diskrit konvensional dengan cara *trial* dan *error*, sehingga diperoleh nilai:

$$\theta_0 = \begin{bmatrix} 6,5460 \\ -8,958 \\ 2,979 \end{bmatrix}$$

dan selanjutnya untuk menentukan inisialisasi matriks kovarian adalah dengan menentukan nilai $\gamma = 2 \times 10^5$. Sehingga diperoleh inisialisasi matriks kovarian $P(0)$ adalah

$$P(0) = \begin{bmatrix} 2 \times 10^5 & 0 & 0 \\ 0 & 2 \times 10^5 & 0 \\ 0 & 0 & 2 \times 10^5 \end{bmatrix}$$

b. Tahap prediksi

Untuk tahap prediksi digunakan persamaan (2.50) sampai (2.53) dengan variabel regresi adalah:

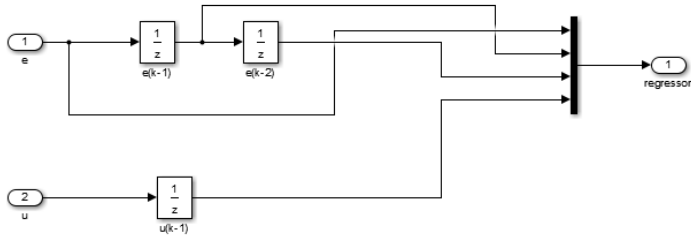
$$\varphi(t) = [e(k) \ e(k-1) \ e(k-2)]$$

sehingga nilai matriks gain, nilai matriks kovarian dan nilai dari estimasi parameter dapat diperoleh dengan menggunakan persamaan:

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + P(t)\varphi(t-1)\varepsilon(t) \\ K(t) &= P(t-1)\varphi(t)(\lambda I + \varphi^T(t)P(t-1)\varphi(t))^{-1} \end{aligned}$$

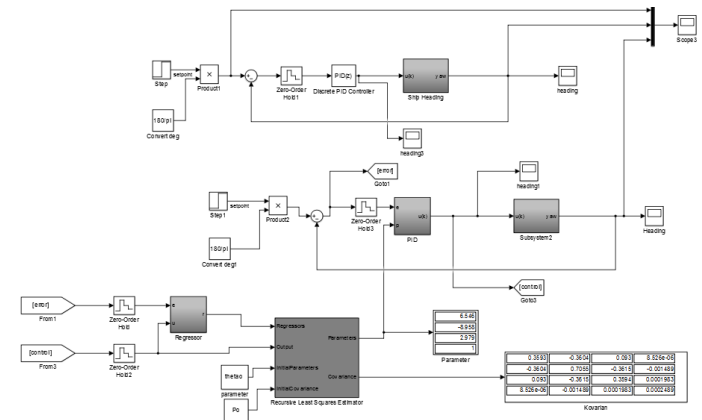
$$P(t) = (I - K(t)\varphi^T(t))P(t-1)/\lambda$$
 dengan nilai *error* prediksi adalah

$$\varepsilon(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1)$$
 perancangan desain variabel *regressor* ditunjukkan pada gambar 4.9.



Gambar 4.9 Desain *regressor*

Selanjutnya, perancangan desain kendali PID adaptif dapat ditunjukkan pada gambar 4.10



Gambar 4.10 Desain PID adaptif

4.4 Simulasi dan Analisis Hasil Perancangan Kendali

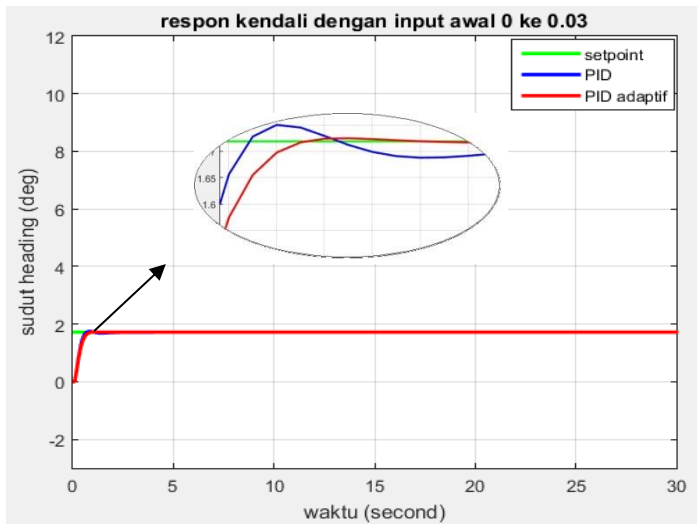
Pada subbab ini akan dilakukan simulasi dari perancangan kendali pada subbab sebelumnya. Pada subbab ini akan dibandingkan simulasi antara pengendali PID dan pengendali PID adaptif. Pengujian dilakukan dengan memberikan *setpoint* dan inisialisasi sudut *yaw* pada pengendali dengan nilai yang bervariasi. Selanjutnya akan dilakukan uji *tracking* dengan memberikan nilai *setpoint* yang berubah-ubah selama waktu tertentu, dan untuk membandingkan kinerja kendali saat meendapatkan gangguan dari luar maka akan dilakukan simulasi dengan menggunakan gangguan impulse, dari hasil simulasi selanjutnya akan dilakukan analisis performansi dari kedua pengendali.

4.4.1 Simulasi dengan nilai *setpoint* berbeda

Pada simulasi ini akan dilakukan pengujian dengan memberikan masukan sinyal *step* dan inisialisasi awal sudut *yaw* yang berbeda selama lima kali. Pada percobaan pertama akan dilakukan simulasi dengan nilai awal sudut *yaw* sebesar 0 rad menuju *setpoint* sebesar 0.03 rad atau 1,7189 deg, percobaan kedua dengan memberikan nilai awal sudut *yaw* 0,03 rad menuju 0,09 rad, percobaan ketiga dengan memberikan nilai awal sudut *yaw* sebesar 0,09 rad menuju 0,18 rad, percobaan keempat dengan memberikan sudut awal *yaw* sebesar -0,03 rad menuju 0,03 rad, percobaan kelima dengan memberikan nilai sudut awal *yaw* -0,03 rad menuju 0 rad. Simulasi dilakukan selama 30 detik dengan parameter PID konvensional yang dimasukan adalah $K_p = 3,311$, $K_i = 5,77$, $K_d = 0,262$. Selanjutnya, hasil simulasi pada kelima percobaan ditunjukkan pada gambar 4.11 sampai gambar 4.15. Selanjutnya dari hasil simulasi akan dihitung rata-rata nilai *settling time* dengan menggunakan kendali PID adaptif dan kendali PID konvensional.

a. Kondisi awal sudut *yaw* 0 rad menuju 0,03 rad

Percobaan yang kedua dengan menentukan nilai awal sudut *yaw* adalah 0 rad menuju *setpoint* 0,03 rad atau 1,7189 deg selama 30 detik, nilai parameter PID konvensional yang dimasukan pada simulasi adalah $K_p = 3,311$, $K_i = 5,77$, $K_d = 0,262$, hasil simulasi ditunjukkan pada gambar 4.11



Gambar 4.11 Simulasi nilai awal 0 rad, *setpoint* 0,03 rad

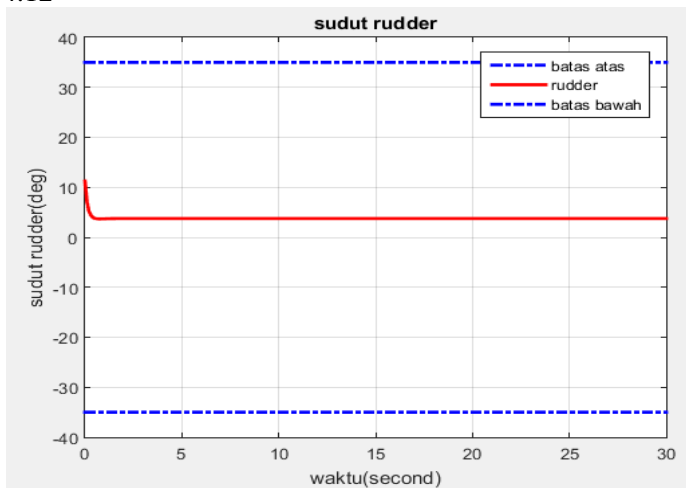
Gambar 4.11 merupakan hasil simulasi respon kendali pada gerak *yaw* dengan posisi awal sudut *yaw* kapal adalah 0 rad dan menuju sudut 0,03 rad atau 1,7189 deg, dari hasil simulasi menunjukkan bahwa pada kendali PID kapal stabil menuju *setpoint* pada waktu ke 2,2 detik sedangkan dengan menggunakan kendali PID adaptif, respon kendali menunjukkan bahwa kapal menuju *setpoint* pada waktu ke 1 detik. Dari hasil simulasi juga terlihat bahwa kendali PID adaptif memiliki *overshoot* lebih kecil dibandingkan kendali PID konvensional, hal ini menunjukkan bahwa

posisi sudut *yaw* dengan menggunakan kendali PID adaptif mampu kembali pada posisi yang diinginkan lebih cepat dibandingkan kendali PID konvensional, selanjutnya perbandingan respon transien dari kendali PID adaptif dan PID konvensional ditunjukkan pada tabel 4.1.

Tabel 4.1 Respon transien nilai awal 0 rad, *setpoint* 0,03 rad

Spesifikasi	PID	PID adaptif
t_d	0,3	0,3
t_p	0,9	1,1
t_r	0,7	1
M_p (%)	1,8%	0,3 %
t_s	2,2	1

Pada saat kapal bermanuver, sudut *heading* dikendalikan oleh sudut *rudder*, pada simulasi ini sudut *rudder* merupakan sinyal kontrol yang dihasilkan dari hasil simulasi, dalam kondisi sebenarnya sudut *rudder* memiliki batas *constraint* yaitu -35 deg sampai 35 deg. Hasil simulasi dari sudut *rudder* ditunjukkan pada gambar 4.12

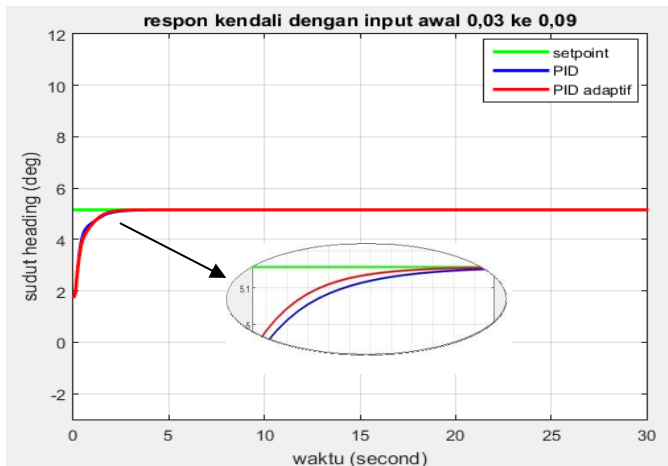


Gambar 4.12 Sudut rudder percobaan 1

Gambar 4.12 merupakan hasil simulasi dari sudut *rudder* pada saat kapal bergerak dari 0 rad menuju 0,03 rad, dari hasil simulasi menunjukkan bahwa sudut *rudder* bergerak dari 11,25 deg menuju 3,749 deg, dengan kata lain untuk menggerakkan kapal memerlukan sudut *rudder* sebesar 11,25 deg, dan pada saat kapal sudah mencapai sudut *yaw* sebesar 0,03 rad sudut *rudder* tidak bergerak dan bertahan pada sudut 3,749 deg. Dari gambar 4.12 dapat dilihat juga bahwa sudut *rudder* tidak melampaui batas *constraint* pada sudut *rudder* kapal.

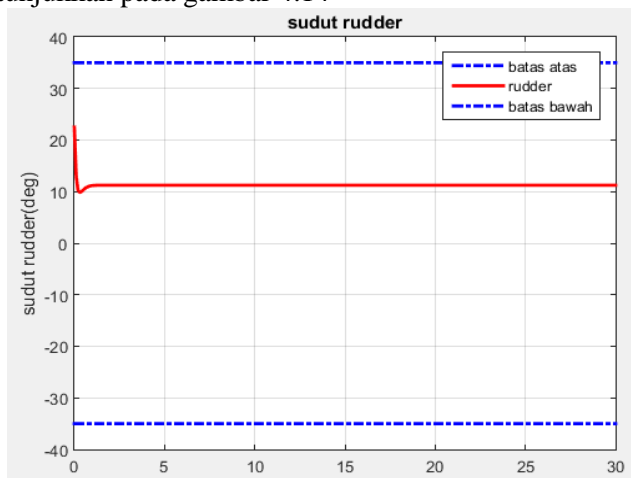
b. Kondisi awal sudut *yaw* 0,03 rad menuju 0,09 rad

Percobaan yang kedua dengan menentukan nilai awal sudut *yaw* adalah 0,03 rad atau 1,7189 deg menuju *setpoint* 0,09 rad atau 5,1566 deg selama 30 detik, nilai parameter PID adaptif dan PID konvensional sama dengan sebelumnya, hasil simulasi ditunjukkan pada gambar 4.12



Gambar 4.13 Simulasi nilai awal 0,03 rad *setpoint* 0,09 rad

Gambar 4.13 merupakan hasil simulasi dengan nilai awal sudut *yaw* adalah 0,03 rad dan menuju *setpoint* 0,09 rad, dari hasil simulasi dapat dilihat bahwa respon kendali PID adaptif menuju *setpoint* lebih cepat dibandingkan PID konvensional, dengan nilai *settling time* kendali PID adaptif adalah 2,5 detik, hal ini menunjukkan bahwa sudut *yaw* kapal stabil menuju *setpoint* pada waktu 2,5 detik. Sedangkan dengan menggunakan PID konvensional nilai *settling time* berada pada waktu 2,9 detik, respon kedua pengendali tidak mengalami *overshoot* hal ini menunjukkan bahwa respon kedua pengendali stabil menuju *setpoint* namun dengan nilai *settling time* yang berbeda. Selanjutnya hasil simulasi dari sudut *rudder* ditunjukkan pada gambar 4.14



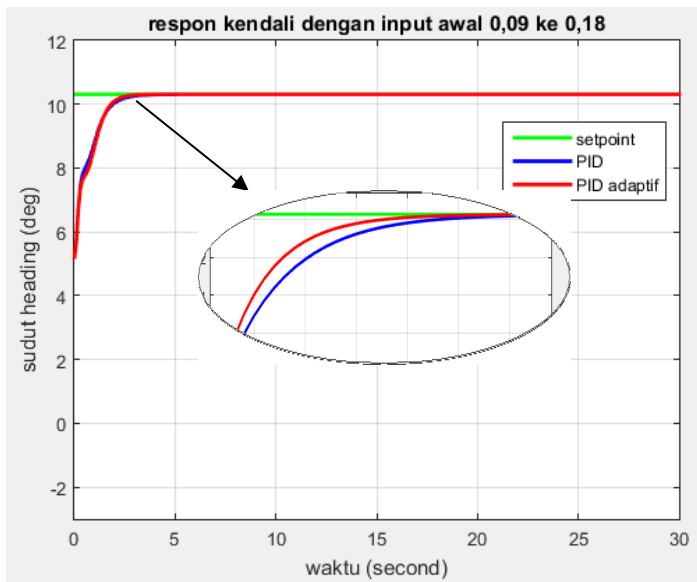
Gambar 4.14 Sudut *rudder* percobaan 2

Gambar 4.14 merupakan hasil simulasi dari sudut *rudder* pada saat kapal bergerak dari 0,03 rad menuju 0,09 rad, dari hasil simulasi menunjukkan bahwa sudut *rudder* bergerak dari 22,5 deg menuju 11,2471 deg, dengan kata lain untuk menggerakkan kapal memerlukan sudut *rudder* sebesar 22,5 deg, dan pada saat kapal sudah mencapai

sudut *yaw* sebesar 0,09 rad sudut *rudder* tidak bergerak dan bertahan pada sudut 11,2471 deg. Dari gambar 4.14 dapat dilihat juga bahwa sudut *rudder* tidak melampaui batas *constraint* pada sudut *rudder* kapal.

c. Kondisi awal sudut *yaw* 0,09 rad menuju 0,18 rad

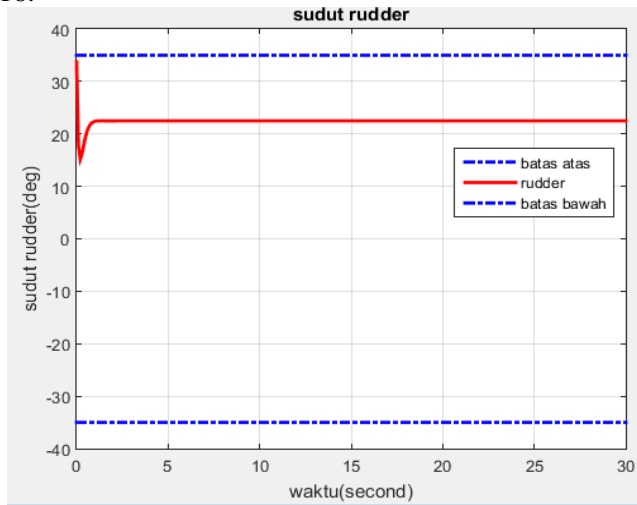
Percobaan yang ketiga dengan menentukan nilai awal sudut *yaw* adalah 0,09 rad atau 5,1566 deg menuju *setpoint* 0,18 rad atau 10,3132 deg selama 30 detik, nilai parameter PID adaptif dan PID konvensional sama dengan percobaan sebelumnya, hasil simulasi ditunjukkan pada gambar 4.15



Gambar 4.15 Simulasi nilai awal 0,09 rad, *setpoint* 0,18 rad

Gambar 4.15 menunjukkan hasil simulasi kendali haluan kapal dengan posisi awal sudut *yaw* adalah 0,09 rad atau 5,1566 deg dan akan menuju *setpoint* 0,4 rad atau 10,3132 deg, dari hasil simulasi dapat dilihat bahwa kedua

sistem tidak mengalami *overshoot*, namun respon kendali PID adaptif mampu menanggapi sistem lebih cepat dibandingkan PID konvensional, desain kendali dengan menggunakan PID adaptif mencapai *settling time* pada waktu 2,9 detik sedangkan PID konvensional mencapai *settling time* pada waktu 3,3 detik, hal ini menunjukkan bahwa dengan menggunakan kendali PID adaptif sudut *yaw* mampu mencapai *setpoint* pada waktu 2,9 detik sedangkan dengan menggunakan PID konvensional sudut *yaw* mencapai *setpoint* pada waktu 3,3 detik. Selanjutnya hasil simulasi dari sudut *rudder* ditunjukkan pada gambar 4.16.



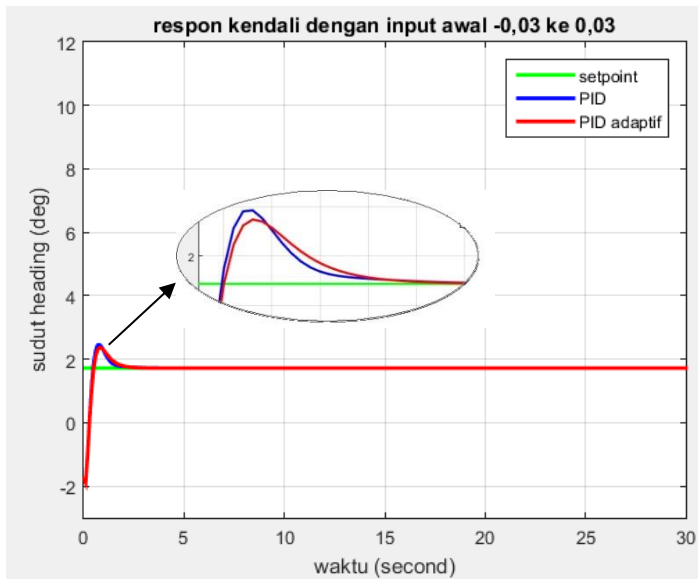
Gambar 4.16 Sudut *rudder* percobaan 3

Gambar 4.16 merupakan hasil simulasi dari sudut *rudder* pada saat kapal bergerak dari 0,09 rad menuju 0,18 rad, dari hasil simulasi menunjukkan bahwa sudut *rudder* bergerak dari 33,752 deg menuju 22,4942 deg, dengan kata lain untuk menggerakkan kapal memerlukan sudut *rudder* sebesar 33,752 deg, dan pada saat kapal sudah mencapai sudut *yaw* sebesar 0,18 rad sudut *rudder* tidak

bergerak dan bertahan pada sudut 22,4942 deg. Dari gambar 4.16 dapat dilihat juga bahwa sudut *rudder* tidak melampaui batas *constraint* pada sudut *rudder* kapal.

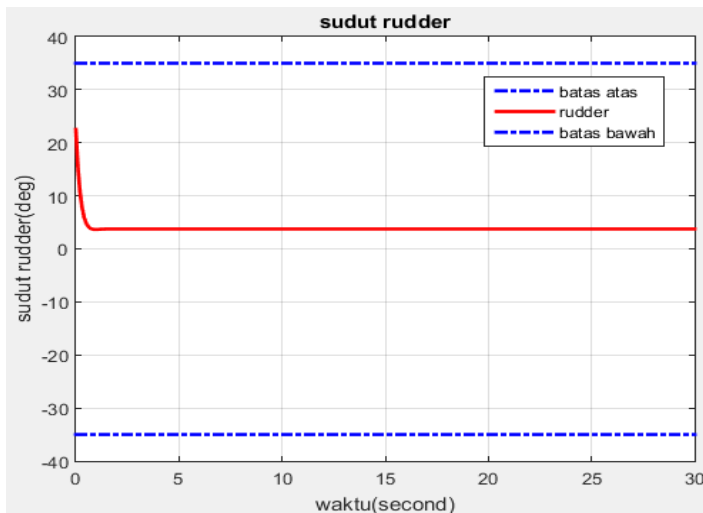
d. Kondisi awal sudut yaw -0,03 rad menuju 0,03 rad

Percobaan keempat dengan memberikan nilai awal sudut yaw sebesar -0,03 rad atau -1,7189 deg menuju *setpoint* 0,03 rad atau 1,7189 deg selama 30 detik. Nilai parameter PID dan PID adaptif sama dengan parameter pada percobaan sebelumnya, tanda negatif menunjukkan posisi dari sudut yaw berada pada arah kiri kapal dan menuju ke kanan sampai sudut 0,03 rad. Hasil simulasi ditunjukkan pada gambar 4.17.



Gambar 4.17 Simulasi nilai awal -0,03 rad, *setpoint* 0,03 rad

Gambar 4.17 merupakan hasil simulasi kendali haluan kapal menggunakan PID adaptif dan PID konvensional dengan percobaan yang diberikan adalah, dengan memberikan nilai awal sudut *yaw* kapal adalah $-0,03$ rad atau $-1,7189$ deg menuju *setpoint* $0,03$ rad tanda negatif menunjukkan bahwa posisi kapal berada pada arah kiri sumbu normal kapal, dari hasil simulasi menunjukkan respon kedua kendali mengalami *overshoot* relatif besar, namun kendali PID konvensional mempunyai *overshoot* lebih besar dibandingkan kendali PID adaptif, jika dilihat pencapaian *settling time* respon kendali PID adaptif mencapai *setpoint* pada waktu 2,6 detik, hal ini menunjukkan bahwa dengan nilai awal sudut *yaw* $-0,1$ rad, dengan menggunakan kendali PID adaptif menuju $0,1$ rad pada waktu ke 2,6 detik, sedangkan dengan menggunakan kendali PID konvensional pencapaian *settling time* pada waktu ke 2,7 detik. Selanjutnya hasil simulasi sut *rudder* ditunjukkan pada gambar 4.18

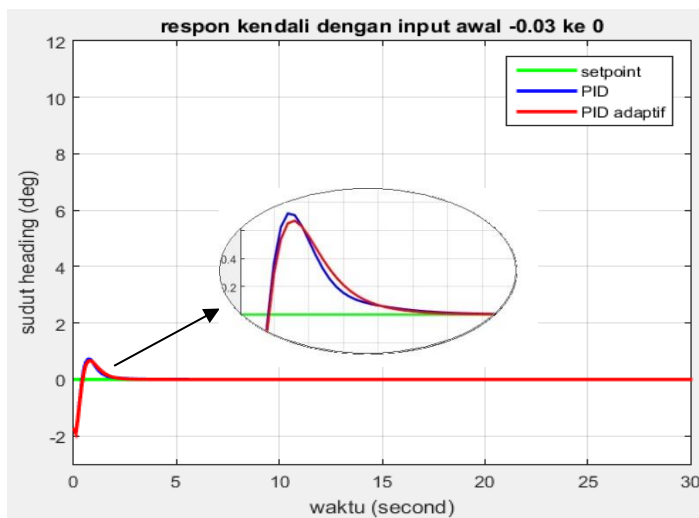


Gambar 4.18 Sudut *rudder* percobaan 4

Gambar 4.18 merupakan hasil simulasi dari sudut *rudder* pada saat kapal bergerak dari $-0,03$ rad menuju $0,03$ rad, dari hasil simulasi menunjukkan bahwa sudut *rudder* bergerak dari $22,5035$ deg menuju $3,749$ deg, dengan kata lain untuk menggerakkan kapal memerlukan sudut *rudder* sebesar $22,5035$ deg, dan pada saat kapal sudah mencapai sudut *yaw* sebesar $0,03$ rad, sudut *rudder* tidak bergerak dan bertahan pada sudut $3,749$ deg. Dari gambar 4.18 dapat dilihat juga bahwa sudut *rudder* tidak melampaui batas *constraint* pada sudut *rudder* kapal.

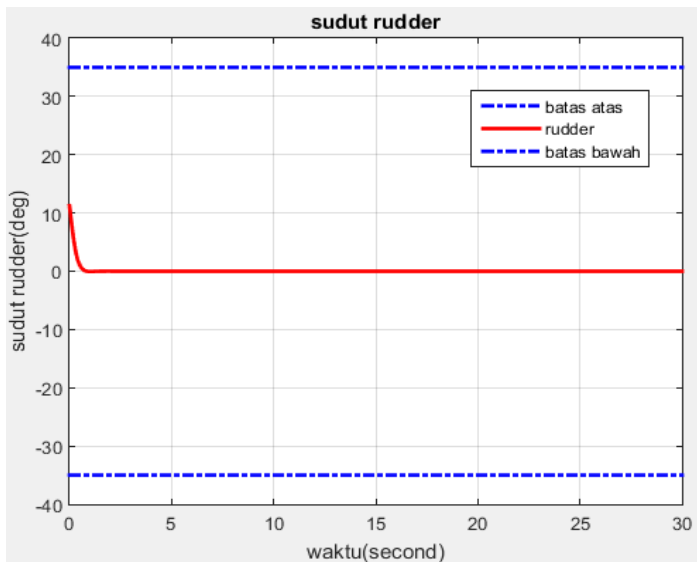
e. Kondisi awal sudut *yaw* $-0,03$ rad menuju 0 rad

Percobaan kelima dengan memberikan nilai awal posisi sudut *yaw* $-0,03$ rad atau $-1,7189$ deg dan akan menuju *setpoint* 0 rad, dengan nilai parameter kendali sama dengan percobaan sebelumnya, hasil simulasi respon kendali PID adaptif dan PID konvensional ditunjukkan pada gambar 4.19.



Gambar 4.19 Simulasi nilai awal $-0,03$ rad, *setpoint* 0 rad

Gambar 4.15 merupakan hasil simulasi kendali haluan kapal menggunakan PID adaptif dan PID konvensional dengan percobaan yang diberikan adalah dengan memberikan nilai awal sudut *yaw* adalah $-0,03$ rad atau $-1,7189$ deg dari sumbu kapal menuju *setpoint* 0 rad. Dari hasil simulasi menunjukkan bahwa respon kedua kendali mengalami *overshoot*, hal ini menunjukkan bahwa respon kendali tidak segera menuju *setpoint* yang diinginkan melainkan mengalami *overshoot* yaitu penambahan sudut *yaw*, namun kedua kendali stabil pada detik ke 2,4 dengan menggunakan PID adaptif dan 2,6 dengan menggunakan PID konvensional dengan kata lain dengan menggunakan kendali PID adaptif mencapai *settling time* pada waktu ke 2,4 detik dan kendali PID konvensional mencapai *settling time* pada waktu ke 2,6 detik. Selanjutnya hasil simulasi dari sudut *rudder* dapat dilihat pada gambar 4.20.



Gambar 4.20 Sudut *rudder* percobaan 5

Gambar 4.20 merupakan hasil simulasi dari sudut *rudder* pada saat kapal bergerak dari $-0,03$ rad menuju 0 rad, dari hasil simulasi menunjukkan bahwa sudut *rudder* bergerak dari $11,2517$ deg menuju 0 deg, dengan kata lain untuk menggerakkan kapal memerlukan sudut *rudder* sebesar $11,2517$ deg, dan pada saat kapal sudah mencapai sudut *yaw* sebesar 0 rad, sudut *rudder* tidak bergerak dan bertahan pada sudut 0 deg. Dari gambar 4.20 dapat dilihat juga bahwa sudut *rudder* tidak melampaui batas *constraint* pada sudut *rudder* kapal.

Selanjutnya dari hasil 5 percobaan akan dihitung rata-rata *settling time* dari kedua pengendali dan selanjutnya akan dibandingkan kinerja dari kedua pengendali. Hasil perbandingan *settling time* ditunjukkan pada tabel 4.2

Tabel 4.2 Perbandingan *settling time*

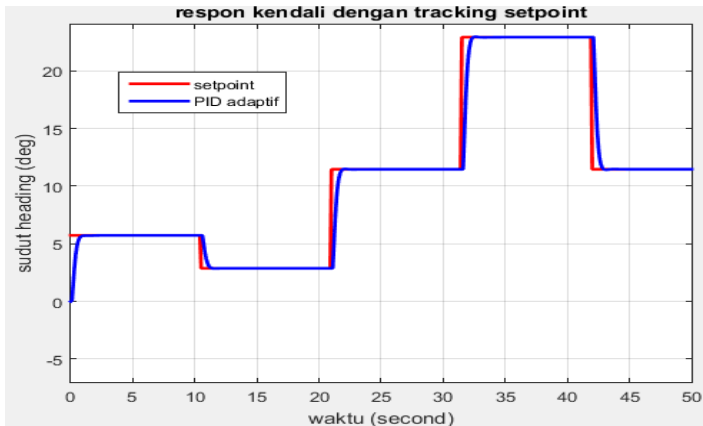
Percobaan	PID	PID adaptif
0 ke $0,03$	2,2	1
$0,03$ ke $0,09$	2,9	2,5
$0,09$ ke $0,18$	3,3	2,9
$-0,03$ ke $0,03$	2,7	2,6
$-0,03$ ke 0	2,6	2,4
Rata-Rata	2,74	2,28

Tabel 4.2 menunjukkan perbandingan hasil simulasi dengan nilai kondisi awal sudut *yaw* dan *setpoint* yang bervariasi diperoleh rata-rata *settling time* dari kendali PID konvensional adalah $2,74$ detik dan PID adaptif $2,28$ detik, sehingga kendali PID adaptif mempunyai respon lebih cepat dibandingkan kendali PID konvensional.

4.4.2 Simulasi dengan *Tracking setpoint*

Pada simulasi uji *tracking* terhadap *setpoint* ditunjukkan respon keluaran sistem terhadap input *reference* atau *setpoint* yang disimulasikan sebagai sinyal

step. Pada uji *tracking setpoint* diharapkan keluaran sistem dapat mengikuti *setpoint* yang diberikan. Untuk mengetahui respon keluaran sistem terhadap perubahan yang bervariasi, perubahan *setpoint* dilakukan lebih dari satu kali, selanjutnya simulasi dilakukan dengan memberikan nilai *setpoint* yang berubah-ubah setiap 10,5 detik selama 50 detik dengan nilai *setpoint* yang diberikan secara berurutan dengan sudut yaw 0,1 rad, 0,05 rad, 0,2 rad, 0,4 rad, dan 0,2 rad. Hasil dari uji *tracking setpoint* ditunjukkan pada gambar 4.21.



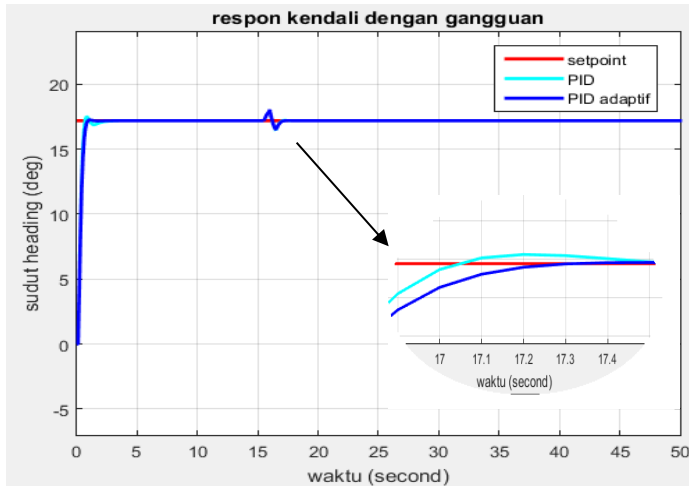
Gambar 4.21 Uji Tracking kapal

Pada gambar 4.21 dapat dilihat bahwa sistem kendali mampu mengikuti nilai *setpoint* lebih dari satu kali dengan baik. Sistem mampu mengubah nilai referensi yang diberikan dengan rata-rata waktu pencapaian *settling time* adalah 1,08 detik.

4.4.3 Simulasi Dengan Gangguan

Simulasi ini dilakukan dengan menambahkan sinyal gangguan yang berasal dari luar sistem, gangguan yang diberikan berupa sinyal impulse yang bersifat sementara, sinyal ini merepresentasikan gangguan yang dialami kapal

pada saat bermanuver, namun hanya sebentar. Pada simulasi ini sinyal impulse yang diberikan berlangsung selama 1 detik pada waktu ke 15,5 detik sampai 16,5 detik. Hasil simulasi respon kendali saat diberikan gangguan ditunjukkan pada Gambar 4.22.



Gambar 4.22 Respon kendali dengan gangguan

Gambar 4.22 merupakan hasil simulasi respon kendali PID adaptif dan PID konvensional dengan memberikan sinyal gangguan berupa sinyal impulse, dari hasil simulasi kendali PID adaptif kembali stabil menuju *setpoint* pada detik ke 17,2, sedangkan kendali PID konvensional mulai stabil menuju *setpoint* pada detik 18,4, dapat dilihat bahwa kendali PID adaptif kembali stabil saat menerima gangguan lebih cepat dibandingkan kendali PID konvensional.

Halaman ini sengaja dikosongkan

BAB V

PENUTUP

Pada bab ini, berisi kesimpulan dari semua kegiatan penelitian dan saran untuk pengembangan dalam Tugas Akhir dan saran untuk penelitian selanjutnya.

5.1 Kesimpulan

Berdasarkan analisis hasil simulasi yang disajikan pada bab sebelumnya, dapat disimpulkan beberapa hal sebagai berikut:

1. Kendali PID adaptif dapat diterapkan pada model kendali haluan Kapal SSV BRP TARLAC (LD-601) karena dari perhitungan sistem telah melalui uji keterkontrolan dan mempunyai nilai $rank M_c = 2$.
2. Dari hasil simulasi menggunakan *software* MATLAB diperoleh hasil sebagai berikut:
 - a. Hasil simulasi pengendali PID adaptif diperoleh karakteristik respon transient sebagai berikut:
$$t_d = 0,3 ;$$
$$t_p = 1,1 ;$$
$$t_r = 1 ;$$
$$M_p = 0,3\% ;$$
$$t_s = 1.$$
 - b. Hasil simulasi dengan percobaan nilai *setpoint* dan kondisi awal sudut *yaw* berbeda menunjukkan bahwa dari 5 percobaan rata-rata *settling time* PID konvensional 2,74 detik sedangkan PID adaptif 2,28 detik.
 - c. Hasil simulasi dengan memberikan gangguan *impulse* menunjukkan bahwa kendali PID adaptif mampu menyesuaikan dan cepat kembali pada *setpoint* dibandingkan PID konvensional.

5.2 Saran

1. Pada Tugas Akhir ini hanya mengendalikan 1 derajat kebebasan kapal, sehingga perlu dikembangkan lagi untuk penelitian dengan mempertimbangkan lebih dari 1 derajat kebebasan.
2. Pada Tugas Akhir ini, gangguan pada sistem hanya berupa simulasi, dapat dikembangkan persamaan gerak kapal dengan mengasumsikan persamaan gangguan yang terjadi pada kapal, seperti persamaan ombak, arus dan angin.

DAFTAR PUSTAKA

- [1] Indrita Hardiana dan Benedicta Trixie. 2014. Menuju Indonesia sebagai Negara Poros Maritim". URL: <http://setkab.go.id/menuju-indonesia-sebagai-negara-poros-maritim/>. Diakses pada tanggal 26 September 2016 .
- [2] Fossen, Thor.I. 2011. **Handbook of Marine Craft Hydrodynamics and Motion Control**. John Willy & Son, Ltd.
- [3] Li, Zhen dan Sun, Jing. 2012. "*Disturbance Compensating Model Predictive Control With Application to Ship Heading Control*". IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, VOL.20, NO. 1.
- [4] Khrisna, M.Murali et al. 2011. "*Theoretical and Experimental Investigation of Gain Scheduling and Adaptive Autopilots For a Model Boat*". International Journal of Engineering Science and Technology (IJEST), Vol.3 No.2.
- [5] Fossen, Thor.I. 1994. **Guidance and Control of Ocean Vehicles** .John Wiley & Sons Ltd.
- [6] Fahmy, Rania. A et al. 2014. "*Adaptive PID Controller Using RLS for SISO Stable and Unstable System*". Hindawi Publishing Corporation, Power Electronica Volume 2014.
- [7] X. Liu, T.Huang, X. Tang, dan H. Xin. 2009. "*Design of self-adaptive PID controller based on least square method*". Proceedings of the 3rd International Conference on Genetic and Evolutionary Computing (WGEC '09), pp. 527–529.
- [8] Lim, C.C dan Forsythe, W. 1983. "*Autopilot for ship control*". IEEE Proceedings, Vol. 130, Pt, D, No.6.

- [9] Amorengen, J. Van. 1984. "*Adaptive Steering of Ships-A model reference Approach*". Automatica, Vol 20. No.1 pp.3-14,1984.
- [10] O'Dwyer, Aidan. 2006. **Handbook of PI and PID Controller Tuning Rules**. Imperial College Press.
- [11] Y. Wakasa, K. Tanaka, and Y. Nishimura,.2012."*Online controller tuning via FRIT and recursive least-squares,*" in Proceedings of the IFAC Conference on Advances in PID Control (PID '12).
- [12] Mudia, Halim. 2012. "Perancangan dan Implementasi Kontroler PID adaptif Pada Pengaturan Kecepatan Motor Induksi Tiga Fasa". Proceeding Seminar Tugas Akhir Jurusan Teknik Elektro FTI-ITS.
- [13] Oktafiani, Nurul Aini. 2014."Perancangan Kontroler PID STR (*self tuning regulator*) direct untuk Mengendalikan Frekuensi Pembangkit Listrik Mini Hidro". Tugas Akhir. Jurusan Teknik Elektro-Fakultas Teknologi Industri, Surabaya.
- [14] Firmansyah, M.Ardi. 2016. "Pemodelan Matematika Kendali Haluan Kapal SSV BRP TARLAC (LD-601) Menggunakan Simulink MATLAB". Laporan Kerja Praktek. Jurusan Matematika ITS.
- [15] Khotimah, Ainun Khusnul. 2016. "Desain Sistem Kendali Proporsional Integral Derivatif (PID) untuk Kestabilan Terbang Gerak Lateral-Direksional LSU-05". Laporan Kerja Praktek. Jurusan Matematika ITS.
- [16] T. Mansour.2011. **PID control, Implementation and Tuning**. InTech.
- [17] Åström, K.J dan Wittenmark, B. 1995. **Adaptive Control**. Addison-Wesley, 2nd edition.

- [18] Subiono. 2013. **Sistem Linear dan Kontrol Optimal**. Surabaya: ITS.
- [19] Ogata, Katsuhiko. 2010. **Modern Control Engineering Fifth Edition**. Prentice Hall.

Halaman ini sengaja dikosongkan

LAMPIRAN A

Source Code

1. Uji keterkontrolan dan Keteramatan

```
clc;
%matriks keadaan dan keluaran
A = [0 1;-8.314185850521662 -
6.079634315639643];
B = [0 ;8.314185850521662];
C =[1 0];
eig(A)
%uji matriks keterkontrolan
k = [B A*B];
rank(k)
%uji keteramatan
o = [C;C*A];
rank(o)
```

2. Input PID adaptif dan plot grafik

```
clc;
%%inisialisasi awal
x0=-0.03;
sp=0;
r = x0*180/pi%%conver to deg
t = sp*180/pi%%conver to deg
time=30;
%%plant sistem
A = [1 0.1;-0.831 0.392];
B = [0;0.381];
C = [1 0];
%%parameter PID diskrit
Kp= 3;
Ki= 5.67;
Kd= 0.2979;
Ts = 0.1;
Td = Kd/Kp;
Ti = Kp/Ki;
```

```

a_1 = Kp*(1+(Ts/Ti)+(Td/Ts))
a_2 = Kp*(-1-(2*Td/Ts))
a_3 = Kp*Td/Ts
%%inisialisasi awal RLS
gamma=2*10^5;
Po = gamma*eye(4); %%matriks kovarian
thetao=[a_1 a_2 a_3 1]; %%parameter awal

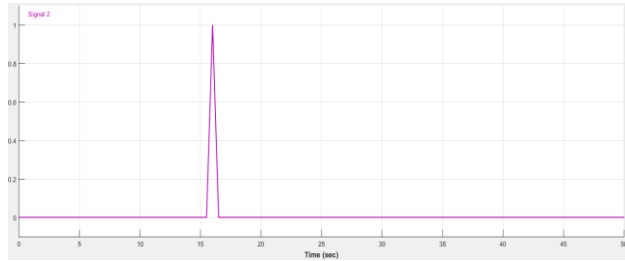
open_system('ujiposisi.mdl');
sim('ujiposisi.mdl',time);
figure(1)%%RLS
x = adaptif(:,1);
y = adaptif(:,2);
z = adaptif(:,3);
n = adaptif(:,4);
plot(x,y,'g-',x,z,'b-',x,n,'r-
','LineWidth',2);
title('respon kendali dengan input awal -
0.03 ke 0');
axis([0 30 -3 12])
legend('setpoint','PID','PID adaptif');
xlabel('waktu (second)');
ylabel ('sudut heading (deg)');
grid on

figure(2)%%sudut rudder
axis([0 30 -36 36])
x = rudder(:,1);
y = rudder(:,2);
z = rudder(:,3);
n = rudder(:,4);
plot(x,y,'b-.',x,z,'r-',x,n,'b-
.','LineWidth',2);
title ('sudut rudder');
legend('batas atas','rudder','batas
bawah');
xlabel('waktu(second)');
ylabel('sudut rudder(deg)');
grid on

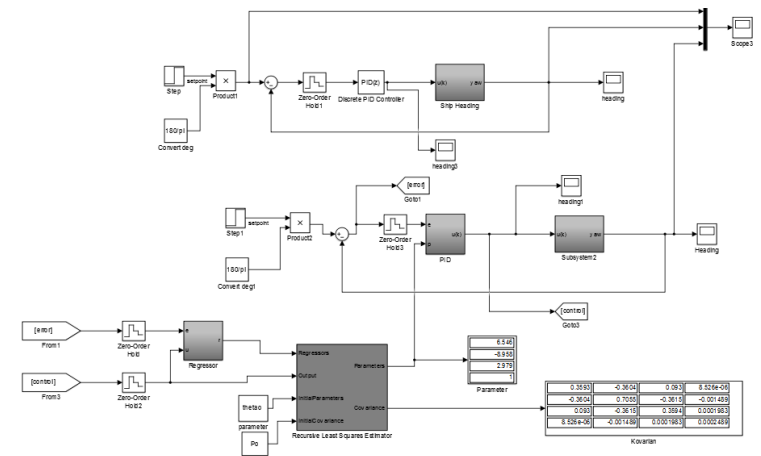
```

LAMPIRAN B DESAIN SIMULINK

a. Sinyal Impulse



b. Desain Kendali PID adaptif



LAMPIRAN C

BIODATA PENULIS



Penulis bernama Moch. Ardi Firmansyah, lahir di Tulungagung, 10 September 1995. Penulis merupakan anak bungsu dari dua bersaudara dari pasangan Bapak Heri Sukarli dan Ibu Mardiyatin. Jenjang pendidikan formal yang ditempuh oleh penulis dimulai dari SDN MAJAN 1 Tulungagung (2001-2007), SMPN 3 Tulungagung (2007-2010), SMKN 3 Boyolangu Tulungagung (2010-2013). Setelah lulus dari SMKN 3 Boyolangu Tulungagung, penulis melanjutkan studi ke jenjang S1 di Jurusan Matematika ITS pada tahun 2013-2017 melalui jalur SBMPTN dengan NRP 1213100073. Di Jurusan Matematika ITS penulis mengambil Bidang Minat Matematika Terapan. Selain aktif kuliah, penulis juga aktif berorganisasi di Himpunan Mahasiswa sebagai Staff PSDM HIMATIKA ITS (2014/2015), Staff PSDM BEM FMIPA ITS (2014/2015), Staff Kaderisasi Ibnu Muqalah (2014/2015), Staff Kaderisasi JMMI ITS (2014/2015), Kepala Departemen PSDM BEM FMIPA ITS (2015/2016) dan mengikuti beberapa kepanitian yang ada di dalam kampus. Selain itu penulis juga melaksanakan Kerja Praktek di PT. PAL INDONESIA (PERSERO) Surabaya pada tahun 2016. Jika ingin memberikan kritik dan saran,serta diskusi terkait Tugas Akhir ini, silahkan menghubungi email:ardifirman23@gmail.com.